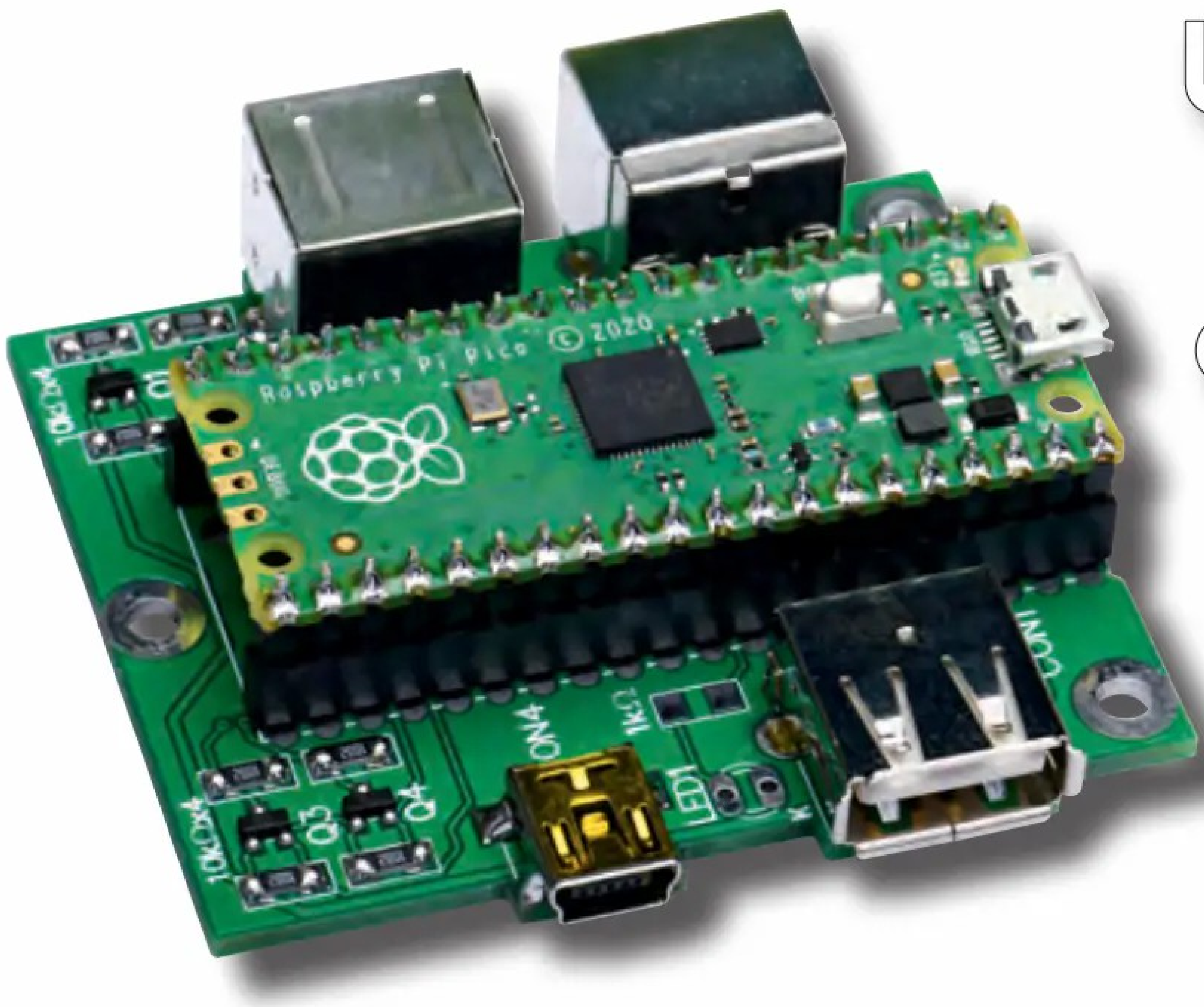
Practical Electronics

The UK's premier electronics, computing and maker magazine

Circuit Surgery
Signal filtering in the digital domain

The History of Electronics
The discoveries behind modern technology

Precision Electronics
How to make precise and repeatable measurements



Use a USB keyboard & mouse with PS/2 compatible devices

A high-quailty microphone preamp for stage & studio work

Finishing the Secure Remote Switch

SECURE REMOTE SWITCH RECEIVER
Power
Relay
(Ack.)
Local

PLUS!

Techno Talk – Low-cost test equipment can be great

Cool Beans – Weird & Wonderful Arduino Projects, part 1

Net Work – The origins of Veroboard (stripboard)

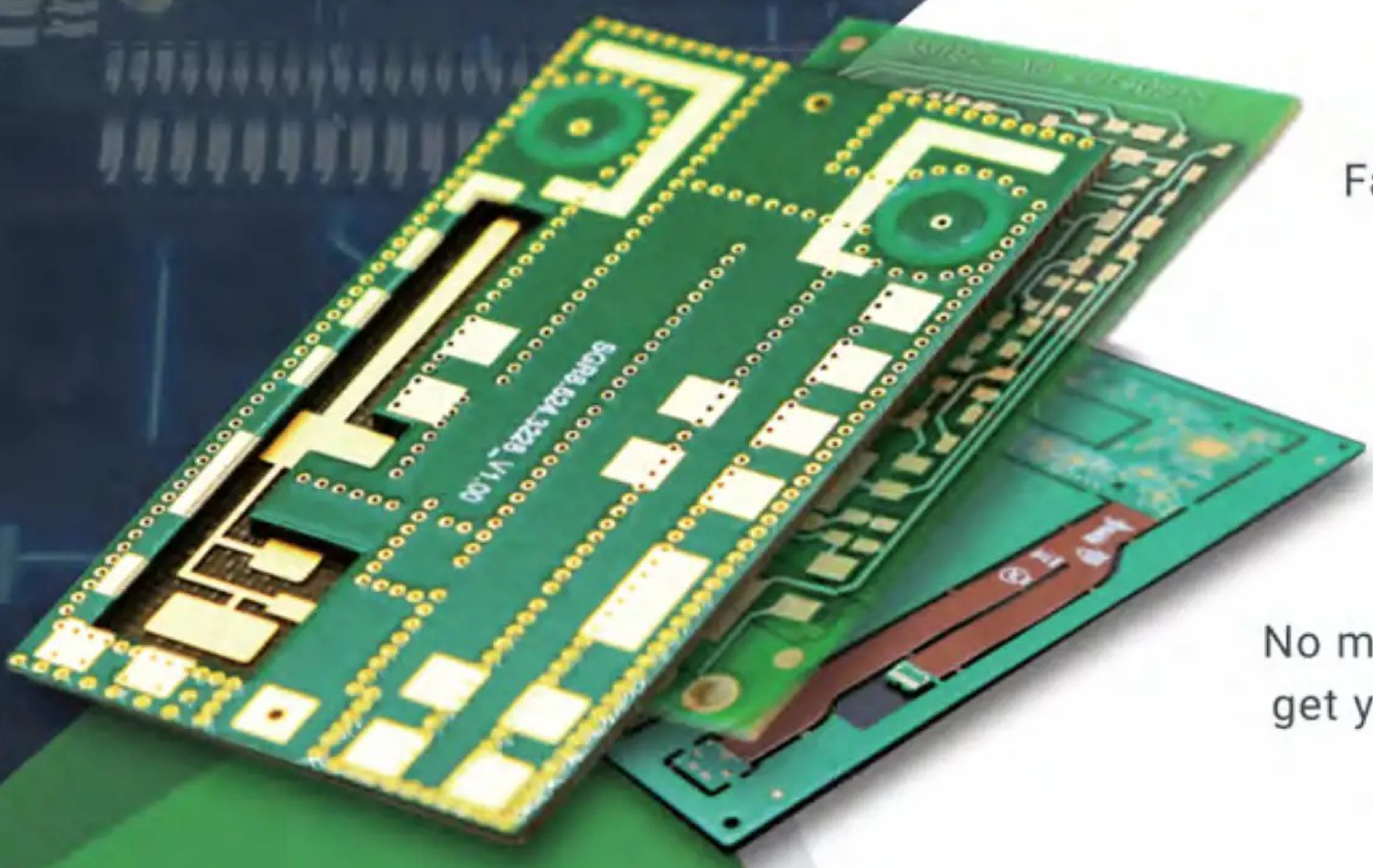
Jan 2025 £5.99

www.electronpublishing.com | @practicalelec | practicalelectronics

PCBWay

EMPOWERING AI INNOVATION, ONE PCB AT A TIME

In the rapidly advancing world of AI, precision and reliability are essential. PCBWay provides high-quality PCB prototyping, assembly, CNC machining, and 3D printing services to bring your AI-driven projects to life. From robotics to smart devices, we offer a complete solution to power your next AI breakthrough.



Precision PCBs

Perfect for the complex designs that drive AI hardware.

Rapid Prototyping

Fast turnaround times as 24 hours to keep your innovations on schedule.

Global Reach

No matter where your lab or workshop is, we'll get your boards to you quickly and affordably.

End-to-End Support

We're more than manufacturers; we sponsor and collaborate with AI developers, startups, and makerspaces worldwide.

Why Choose Us?

With PCBWay as your partner, you can focus on pushing the boundaries of AI, while we ensure your hardware is up to the challenge.

Start building the future of AI today!

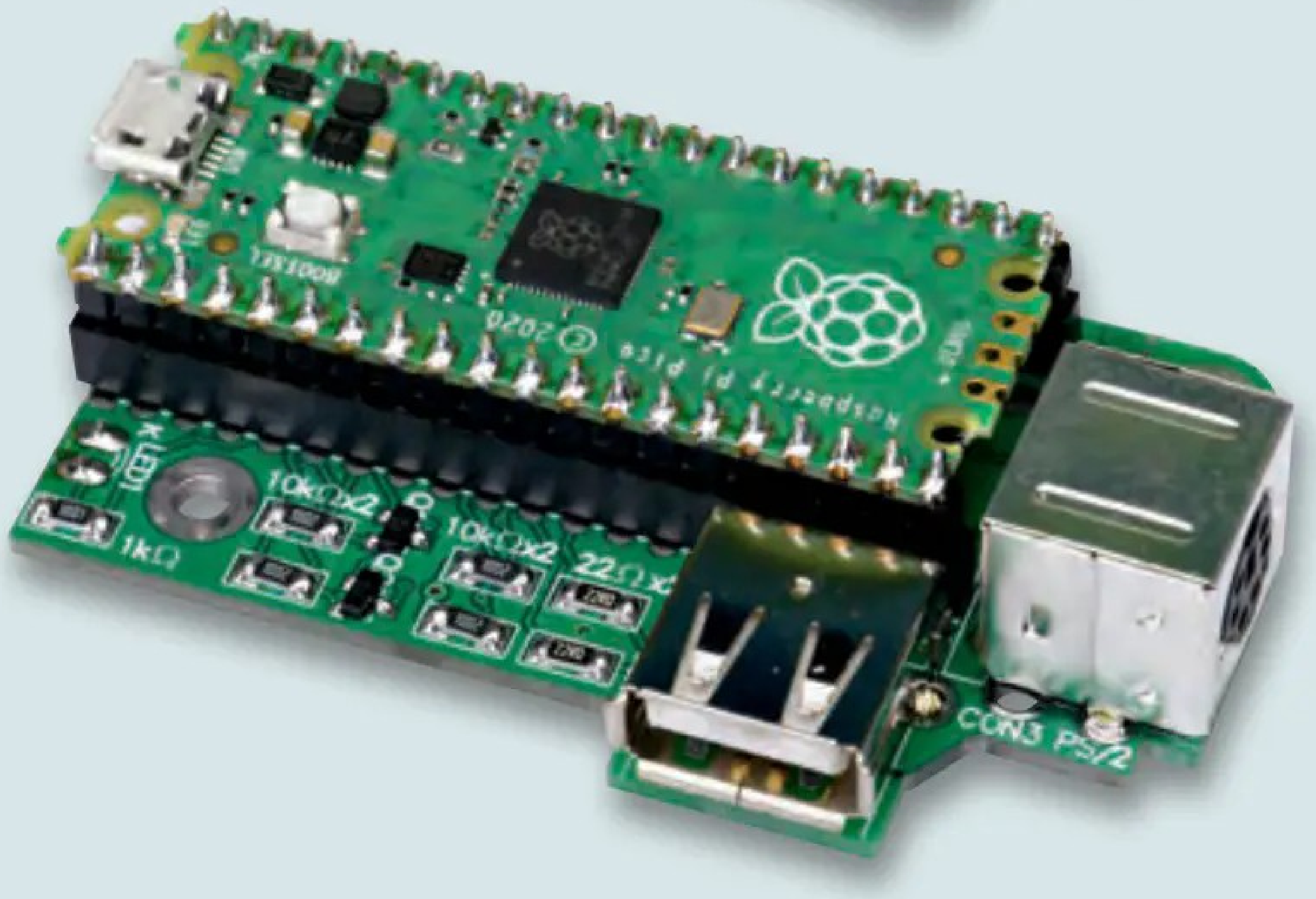
Visit PCBWay.com to discover how our PCBs can power your next AI innovation.

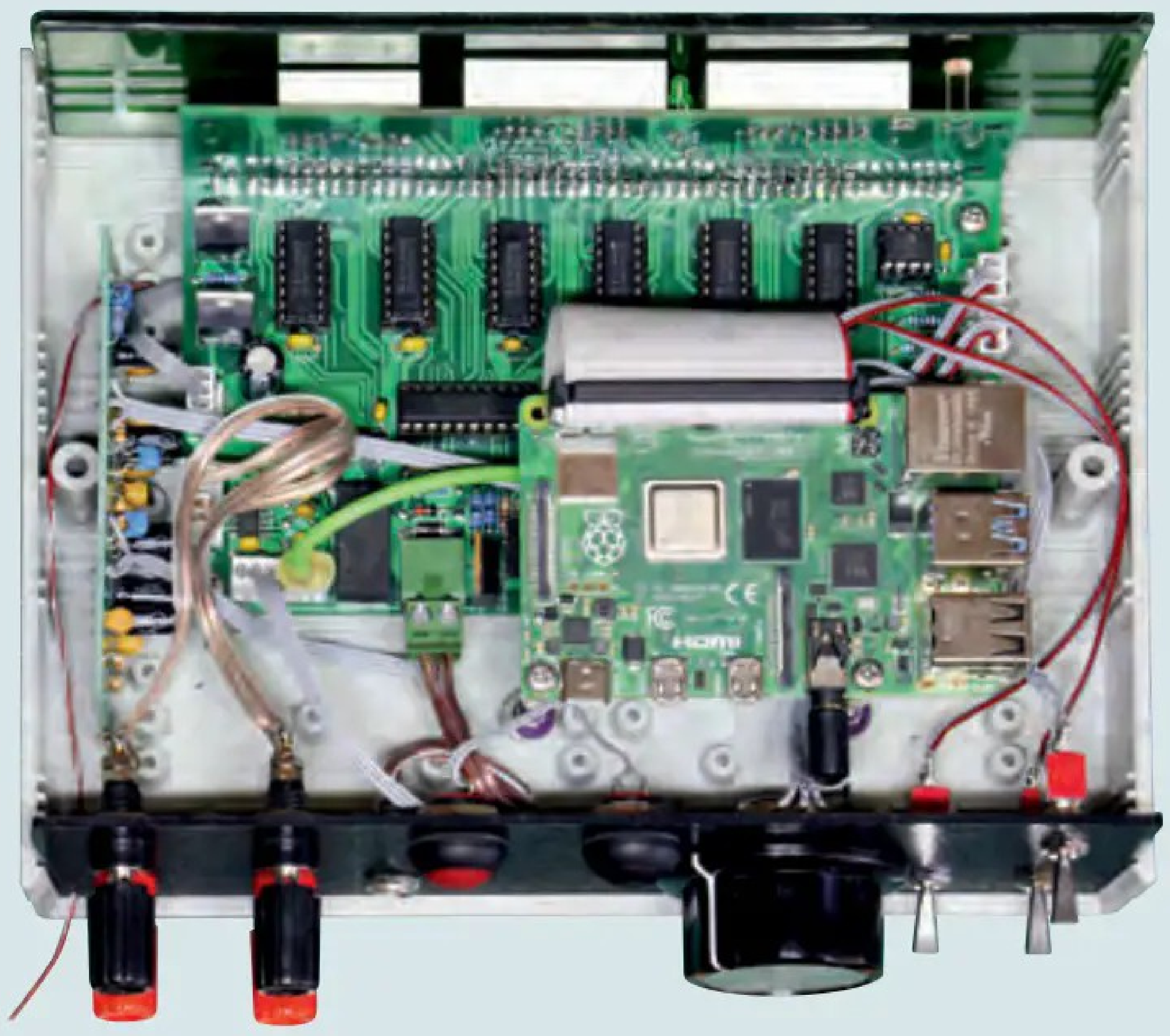
service@pcbway.com
www.pcbway.com

Practical Electronics

Volume 54. No. 1
January 2025
ISSN 2632 573X

Contents





Projects and Circuits

High-quality Microphone Preamplifier by Phil Prosser 14
This small box runs from a plugpack and offers a flat frequency response, very low distortion, low noise and adjustable gain. It fits in a very rugged diecast aluminium enclosure and is great for both stage and studio work!

USB to PS/2 Keyboard & Mouser Adaptors by Tim Blythman 42
Because PS/2 is very easy to interface with a microcontroller, kit computers like the VGA PicoMite include a PS/2 keyboard interface. This Adaptor allows you to connect a USB keyboard or mouse to a unit that expects equivalent PS/2 devices.

Raspberry Pi-based Clock Radio, part two by Stefan Keller-Tuberg 52
This new clock radio design can act as a media player and will synchronise alarms between multiple clocks. Last month we described how it works. Now it's time to get into the construction process. We'll also explain how to use its features.

Secure Remote Mains Switch, part two by John Clarke 70
Last month, we introduced the new Remote Mains Switch, which uses rolling codes for security and supports up to 16 keyfob transmitter per receiver. In this issue, we have all the construction, testing and set-up instructions.

Series, Features and Columns

The Fox Report by Barry Fox 3
How to avoid being fobbed off by an AI bot

Net Work by Alan Winstanley 4
The origins of Veroboard (stripboard), a product that transformed our hobby, Ryobi portable soldering irons and using tool batteries to power other devices.

Max's Cool Beans by Max The Magnificent 7
Weird & Wonderful Arduino Projects, part one: DIP LED bar graph displays

The History of Electronics, part one by Dr David Maddison 26
We wouldn't have computers, smartphones or artificial intelligence without the work of thousands of people who created the foundations of modern electronics.

Circuit Surgery by Ian Bell 33
Signal filtering in the digital domain

Techno Talk by Max The Magnificent 38
Marvelling at how good inexpensive test equipment is these days

The Quason VL6180X laser rangefinder module by Jim Rowe 39
This module uses infrared (IR) light to accurately sense the proximity of objects to well over 100mm. It's based on FlightSense technology by ST Microelectronics.

Precision Electronics, part one by Andrew Levido 64
The first article in a series about precision electronics design. It covers a range of topics and uses real examples and real components to demonstrate the concepts.

Regulars and Services

Editorial 2
Two new series for the magazine

Reader services – Editorial and Advertising Departments 2

Exclusive Microchip reader offer 25
Win a Microchip Explorer 16/32 Development Kit

Subscribe to Practical Electronics and save money 63

Practical Electronics PCB Service 76
PCBs for Practical Electronics projects

Classified ads and Advertiser index 79

Next month! – highlights of our next issue of Practical Electronics 80

Made in Australia & the UK.

Written in Britain, Australia, the USA and Ireland.

Read everywhere.

© Silicon Chip Publications Pty Ltd 2024

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The February 2025 issue of *Practical Electronics* will be published on Friday, 3rd of January 2025 (see p80).

Renewals paid by PayPal, Visa and Mastercard are now handled via our website:
www.electronpublishing.com

Practical Electronics

Volume 54. No. 1
January 2025
ISSN 2632 573X

Editorial

Editorial offices
Electron Publishing (Aust.)
Unit 1, 234 Harbord Road
Brookvale NSW 2100
Australia
Tel +61 2 9939 3295
Email pe@pemag.au
Web www.electronpublishing.com

Address mail to:
Electron Publishing (Australia)
PO Box 194, Matraville NSW 2036
Australia

Advertising enquiries
+61 2 9939 3295 pe@pemag.au

Editor	Nicholas Vinen
Publisher	Nicholas Vinen
Digital subscriptions	Stewart Kearn **Tel** 01202 880299
Online Editor	Alan Winstanley
Web Systems	Kris Thain
Production	Bao Smith
Technical staff	Tim Blythman, John Clarke

Print subscriptions
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom
Tel 01202 087631
Email pesubs@selectps.com

Technical enquiries
We regret that technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine.

Questions about articles or projects should be sent to the editor by email: pe@pemag.au

Projects and circuits
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

Some projects and circuits published in *Practical Electronics* employ voltages that can be lethal. Do not build, test, modify or fix any mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies
Silicon Chip Publications may offer kits or other parts for making our projects, but not in all cases. When kits are not available, readers will need to find and source parts themselves. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Two new series for the magazine

I mentioned last month that I was working on getting UK-based authors of previous popular series back into the magazine and that they would hopefully resume by the middle of 2025.

In the meantime, we have a couple of new series that I think *PE* readers will find quite interesting – I do, anyway.

The first is a six-part series on the people who laid the groundwork for electronic theory and their discoveries. We created this series because we realised that there was so much theory behind what we do that we don't often think or write about, but it is important nonetheless.

You will likely have heard the names of many people mentioned in the articles: Huygens, Galvani, Coulomb, Volta, Laplace, Dalton, Seebeck, Ampère, Gauss, Ohm, Faraday, Morse, Henry, Weber, von Siemens, Joule, Bequerel, Lord Kelvin, Kirchhoff, Maxwell, Crookes, Röntgen, Bell, Edison, Weston, Tesla, Marconi, Einstein, Yagi, Baird and Armstrong to name a few. But there are far more you likely won't know of.

The second is an educational series on Precision Electronics that will likely end up with nine parts, although that could vary. It contains a mix of practical knowledge and light theory that helps to explain how to go about designing circuits for precision measurement. You will find it somewhat similar to *Circuit Surgery* or *Audio Out* but on a different topic.

One thing I particularly like about this series is that he performs some calculations to show what sort of precision we can expect from various circuit configurations, then actually builds those circuits and measures them. So you find out exactly how well the theory matches up to the reality by the end of each instalment.

The author of that series has also designed major projects for us in the past and has a few good ones coming up that will likely be published later this year. He also designed circuits professionally for a number of years. It's great to have people with such hands-on backgrounds writing articles for the magazine.

Nicholas Vinen, Electron Publishing (Australia)*
Publisher & Editor,
Practical Electronics Magazine

* a division of Silicon Chip Publications Pty Ltd.

The Fox Report

Barry Fox's technology column

How to avoid being fobbed off by an AI bot

Every day, it seems we get a new Artificial Intelligence system such as ChatGPT from OpenAI, Copilot from Microsoft, Gemini from Google or Meta AI from the owner of Facebook, Instagram, WhatsApp and Messenger. Each variant is hailed as the currently best or upcoming solution.

AI used to be all about the Large Language Models (LLMs), on which tools like ChatGPT depend. Now we also have Small Language Models (SLMs), which are, err, smaller that LLMs.

LLMs are computer machines (in the way Alan Turing used the word, rather than mechanical engines) that can understand and create human language text. They are 'trained' by analysing vast amounts of raw data (text, images, calculations) and then use what they have learned to handle a wide range of 'mental' tasks.

LLMs understand questions in any language and concoct answers based on past information. They can translate to and from foreign languages, summarise big documents in a few words and write essays and reports. They are also increasingly used to replace humans in customer care roles.

Sometimes the service system makes it clear that callers are dealing with a chatbot (they are given 'friendly' robot names like Cora or Clara). However, other times, the service tries to fool callers into thinking that they are talking to a human.

SLMs are smaller because they are more specialised and trained to do specific jobs. They can, for instance, focus on healthcare or specialist technical help. For example, telling a patient what pills to take, or advising an engineer on what screw to turn and how far to tighten it. That can be much easier and quicker than poring through complex manuals, trying to decipher the prose and figure out what to do in this particular situation.

Nvidia (who became huge on the back of making 3D rendering engines for gaming) is getting richer by the minute. That's because it invested in the design of chips that all forms of AI depend on. The new 'Blackwell' chips are expected to cost around US$40k (~£30,000) each.

Meanwhile, more than 13,000 household-name authors, musicians, and actors have signed a joint statement on AI training which declares, "The unlicensed use of creative works for training generative AI is a major, unjust threat to the livelihoods of the people behind those works, and must not be permitted" (https://pemag.au/link/ac27).

The gist of the complaint is that AI model-building needs raw data to train the system, which is obtained by 'scraping' it from the Internet, and no-one is yet paying a fair price for that data. Bodies behind the complaint, such as Fairly Trained, the non-profit group set up by UK composer Ed Newton-Rex, want world-wide agreement on a way to get payment for the scraped data. They also want to block the scraping of data unless payments are made.

That will not be easy, and it will be even less easy to control the future use of data that has already been scraped and has already been used to train all the AI systems that are already available.

Still, surely it is possible to implement such a system. Google is already offering a watermarking system that lets exam boards and business employers check for text that has been generated by Gemini AI.

For what it's worth, I have now temporarily given up on trying every new AI system that clever friends tell me is definitely now the best. I don't yet trust any of them not to try to be helpful (or save machine face) by making up what they don't really know. For now, I just stick with basic Google searching, which is essentially using a sorted index of the Internet.

I give Google a few carefully chosen keywords and ignore all the obvious paid-for results and clickbait, while sifting through the mixed bag of what's left, judging its relevance to me based on my personal experience and doing some separate searches for cross-checking. This takes longer than asking an AI system a complicated question, but I don't want any 'thing' making my relevance judgements. Next year, perhaps.

Likewise, I try to avoid company chatbot helplines because they are usually a pain. But avoidance gets harder all the time. So far, it is typically still possible to deal with such bots by carefully drafting a detailed question off-line and then copy-and-pasting it into the query box. Sometimes, that alone may produce a useful reply.

More often, it's necessary to go on pasting the same pre-prepared question in every box space offered until the bot gives up and passes the query to a human. We then only have to hope that the human can help (if they're only allowed to work from scripts, they're about as much use as a bot…).

This AI-avoidance tactic is the modern equivalent of repeatedly keying star or hash codes into an automated phone line until an operator jumps in. It is obviously time-wasting. Here's a challenge for readers: how best to draft an online query or email that an AI help bot immediately knows it can't handle and needs to refer to a human?

Here's my early attempt at bypassing AI – successfully tested recently during correspondence with a supermarket about rotten food. Attach or embed an image that contains a key fact of the matter. For instance, refer in the text body to a scanned image of a receipt. The worse the scan quality the better, as long as it is legible to the human eye.

Then refer in the text body to key information contained only in the receipt, such as amount paid, product description and date of payment – without including that information as text.

Of course, it is possible to extract text from an image, but optical character recognition (OCR) needs to be initiated, and the result interpreted and correlated with the main body text. OCR can fail with low-quality images (one of the facts those incredibly annoying "Captchas" rely on).

For the time being, at least, it will be easier just to get a human to read the text and look at the image.

This is another good reason for sometimes putting complaints to a company in a snail mail letter (now that faxing is no longer the easier option). Enclose a hard-copy of a receipt with the letter and refer to it in the text. To automate a reply, the company will have to scan and OCR both the letter and receipt, and correlate the two. It will very likely just be easier for a real, live human to read and reply.

Net Work

Alan Winstanley

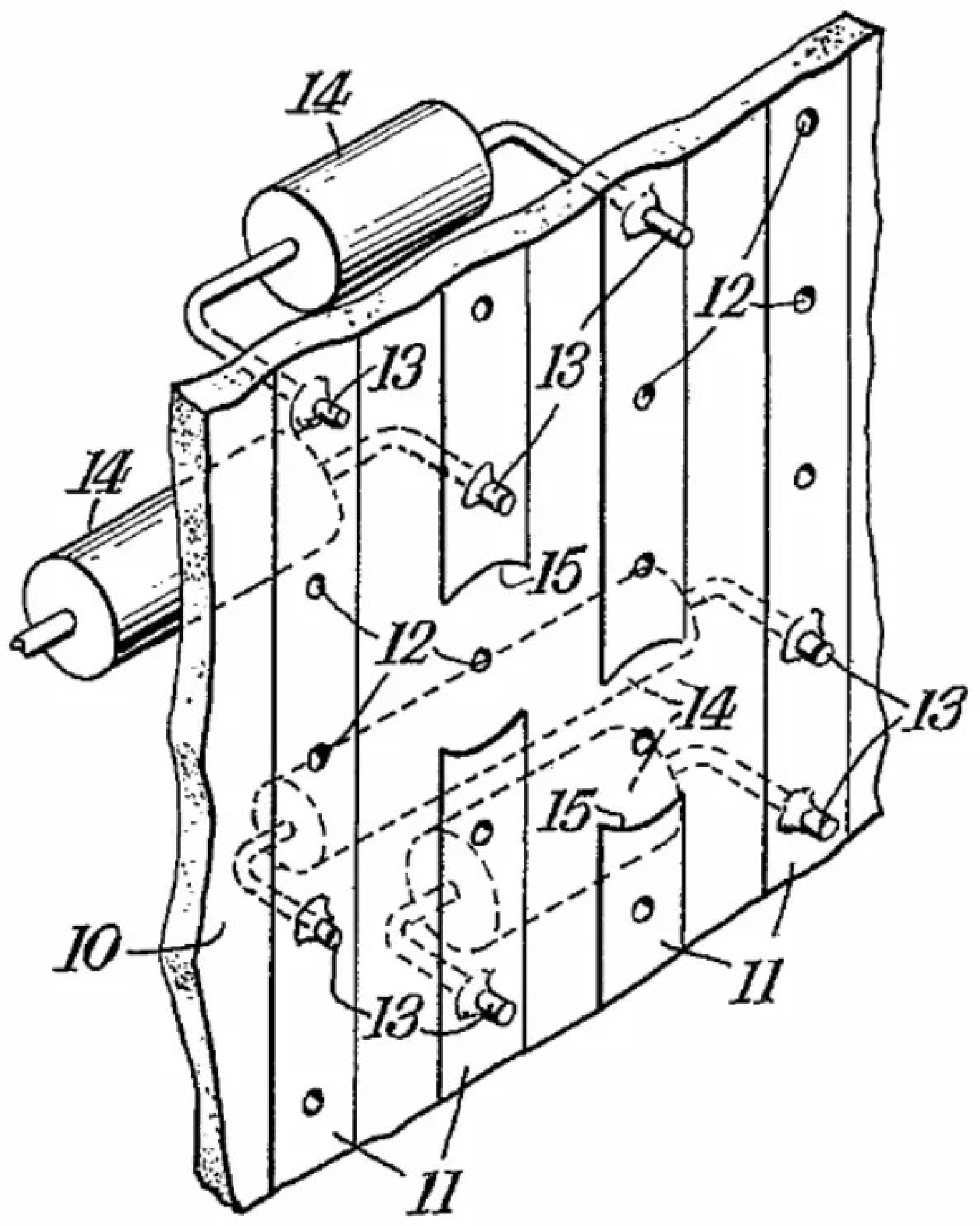
Concluding our nostalgic 60-year journey back to the origins of Practical Electronics, this month's Net Work reminisces about a product that transformed our hobby, and made electronics construction accessible to every enthusiast.

In my previous two columns, I looked back at the 60-year history of *Practical Electronics*. The first issue appeared in late 1964. It was designed to meet the demands for hobby projects that exploited new and rapidly evolving semiconductors.

As vacuum tubes were swept aside and transistors arrived on the hobby scene, new low-voltage projects would be built using unwieldy tag strips or 'turret boards', although some projects utilised basic printed wiring boards (now called printed circuit boards or PCBs) instead.

I noticed in some 1966 back issues a multi-purpose project PCB called the "Bonanza Board" which, cleverly, could carry a dozen different circuits. The home constructor had to etch a board for each project themselves, using quite a perilous process.

An etch resist pattern was traced by hand (or offset using carbon paper) onto copper-clad phenolic panels, and the desired pattern was then 'inked in' using lacquer or cellulose paint. The board was then immersed in a noxious pickle of hydrochloric acid and iron perchloride, better known as ferric chloride, before being drilled.



Vero's original engineering drawing taken from the 1959 patent, showing how copper strips could be linked or drilled out to form a working circuit.

Understandably, this hazardous and laborious process was not for every hobbyist, so a method of easily building smaller, one-off circuits or prototypes was sought. A system that also allowed 'chopping and changing' when perfecting low-voltage circuits would also be ideal for experimenters.

The answer came, of course, in the form of Veroboard, the first 'stripboard' product of its kind, dedicated to the rapid prototyping and low-volume production of electronic circuits.

Enter the Matrix

The story of Veroboard turned out to have many twists and turns. With the help of the UK Intellectual Property Office, I obtained copies of the original 1959 UK and French patents that were submitted by staff at Vero Precision Engineering Ltd (VPE) at the time.

The re-discovered drawings for "Improvements in Wiring Boards" illustrated an idea that would become familiar to us all. Veroboard was described as a regular matrix of holes punched through non-conductive SRBP (synthetic resin bonded paper) panels, with copper strips on the underside that could be configured by linking or interrupting them.

Various references describe the firm's origins. Geoffrey Verdon-Roe (1918–1997) was the Managing Director of VPE in the late 1950s. The firm was claimed to be a very highly respected British precision machine-tool manufacturer, but was not a company that had any connection with electronics.

The Roe family name will be familiar to many, however. Geoffrey's father was (Edwin) Alliot Verdon Roe (1877–1958), who pioneered and co-founded the A. V. Roe (Avro) aircraft company in 1910. He's commemorated by a local authority plaque in London; see https://www.londonremembers.com/memorials/alliott-verdon-roe-sw18

Despite Roe having some controversial political affiliations, ultimately Avro designed 70 notable British aircraft, including the famous WW1 Avro 504 biplane, the Lancaster bomber, the Shackleton, and the Avro Vulcan, each arguably the finest aircraft of its type at the time. (I'm reminded of Sydney and Frederick Camm at this point – see last month's Net Work).

Roe later hyphenated his surname to Verdon-Roe. Avro was swallowed up by Hawker Siddeley in 1962, which itself is now part of the BAE Systems aerospace company.

The suggestion is that VPE then recruited some technical staff from Saunders-Roe (SARO) with the aim of developing precision electronic controls for their machine tools, something they ultimately failed to achieve.

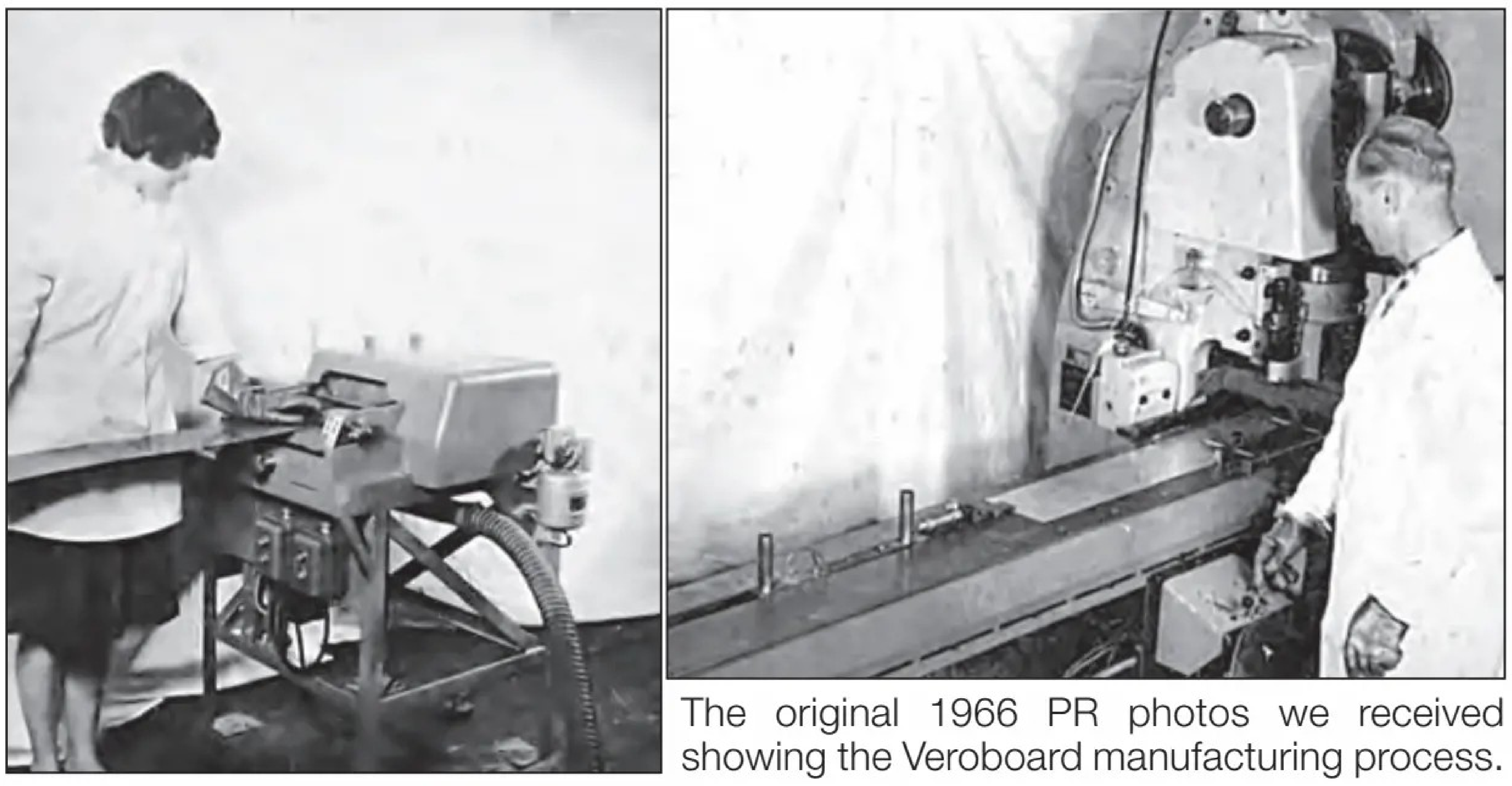
The former S. E. Saunders Ltd was a flying-boat manufacturer in which Alliot Verdon-Roe had previously co-invested, if not rescued from financial ruin altogether. He was also chairman of Vero Precision Engineering until his death in 1958.

SARO was also subcontracted to work on Britain's "Black Knight" (later, Blue Streak) space rocket project, which was test-launched successfully from Woomera, Australia in 1958 (see https://youtu.be/Mnvd3P9H3o40). For anyone interested, YouTube has plenty more resources including a video at https://youtu.be/Ne8shDmcd5g and a TV documentary at https://youtu.be/oGOTjAL5VVg

I can only speculate that in the late 1950s, Britain's aerospace industry possibly sought an electronic circuit prototyping system fit for the transistor era, which sparked the quest for a solution. Joining the dots together, this might have led to Vero's now semi-redundant electronics team being tasked with developing a product suitable for the electronics industry.

Their 1959 patent for an "improved wiring board" was duly published in 1961. The patent also suggested using holes in other patterns (eg, circles) and possibly using conductors other than copper, or even etching copper strips using printed circuit techniques instead. These ideas would hopefully prevent piracy or adaptations of their ideas by others.

Manufacturing the new stripboard



The original 1966 PR photos we received showing the Veroboard manufacturing process.

demanded expertise in precision engineering, including indexing, stepping, punching and milling. VPE had the necessary machine tools and skills, but once the design and production had been perfected, the manufacture and further development of Veroboard was hived off into a separate electronics company.

One of the patent's authors, Terence Fitzpatrick, was then unceremoniously sacked by Geoffrey Verdon-Roe in January 1961, his electronic designs skills no longer being needed.

Craft work

In the December 1966 issue, we featured a news item showing Veroboard being manufactured by Vero Electronics at their new factory in Chandlers Ford, Eastleigh, in Hampshire. It showed an indexing machine piercing sheets of copper-clad board (probably 0.2-inch [5.08mm] or 0.15-inch [3.81mm] matrix at the time).

They perhaps used a vacuum extraction system to remove debris (those holes must go somewhere, after all), followed by a second process using milling cutters to remove unwanted copper between the strips. Judging from the photos, production looked like a slow process.

Practical Electronics and amateur constructors have long had an affinity for Veroboard. It would be fair to say that its role was critical in enabling hobby electronics to blossom, especially with the digital age coming just over the horizon in the 1970s.

Veroboard could carry dual-in-line chips admirably, and constructors were adept at linking copper strips with jumper wires, or isolating them using a special 'spot face cutter' tool [a drill bit of the right size would work too – Editor].

Although Veroboard was aimed at electronics professionals, happily, they decided to offer four sizes for the hobby trade. Vero Electronics even manufactured a sample size for *Practical Electronics* readers. It was 10 strips x 24 holes, a format I became very familiar with during the late 1970s, when Editor Fred Bennett asked me for some simple circuits that novices could build using it.

My "Uniboards" series was duly published in *Everyday Electronics*. I guess Uniboards was inspired by the "Bonanza Boards" idea mentioned earlier, only an infinite number of circuits could now be built by anybody, thanks to the adoption of this universal stripboard.

The first edition of *Everyday Electronics* in 1971 had a free sample of Veroboard (as shown in November 2024's Net Work), and the October 1981 issue carried not only some free Veroboard but also a full-page advert from Vero Electronics themselves, celebrating 20 years of the eponymous product.

Vero wrote that "such is the influence that Veroboard has had, and is still having, that the very name has become part of general vocabulary." Ten simple projects were published in the 1981 issue, including three Uniboards designs.

Computer World

Photographic techniques using UV-sensitive chemicals and developers gradually emerged, and by the late 1980s, professional PC software enabled engineers to design complex PCBs on-screen.

Hobby electronics tends to follow industrial trends and design and construction techniques have since changed out of all recognition. The use of surface-mount devices and microcontrollers is widespread and wholly different skill sets are needed to utilise them successfully. There's no need to make boards from scratch either, as PCB designs can be sent to China (or Europe) via the web and delivered just a few days later.

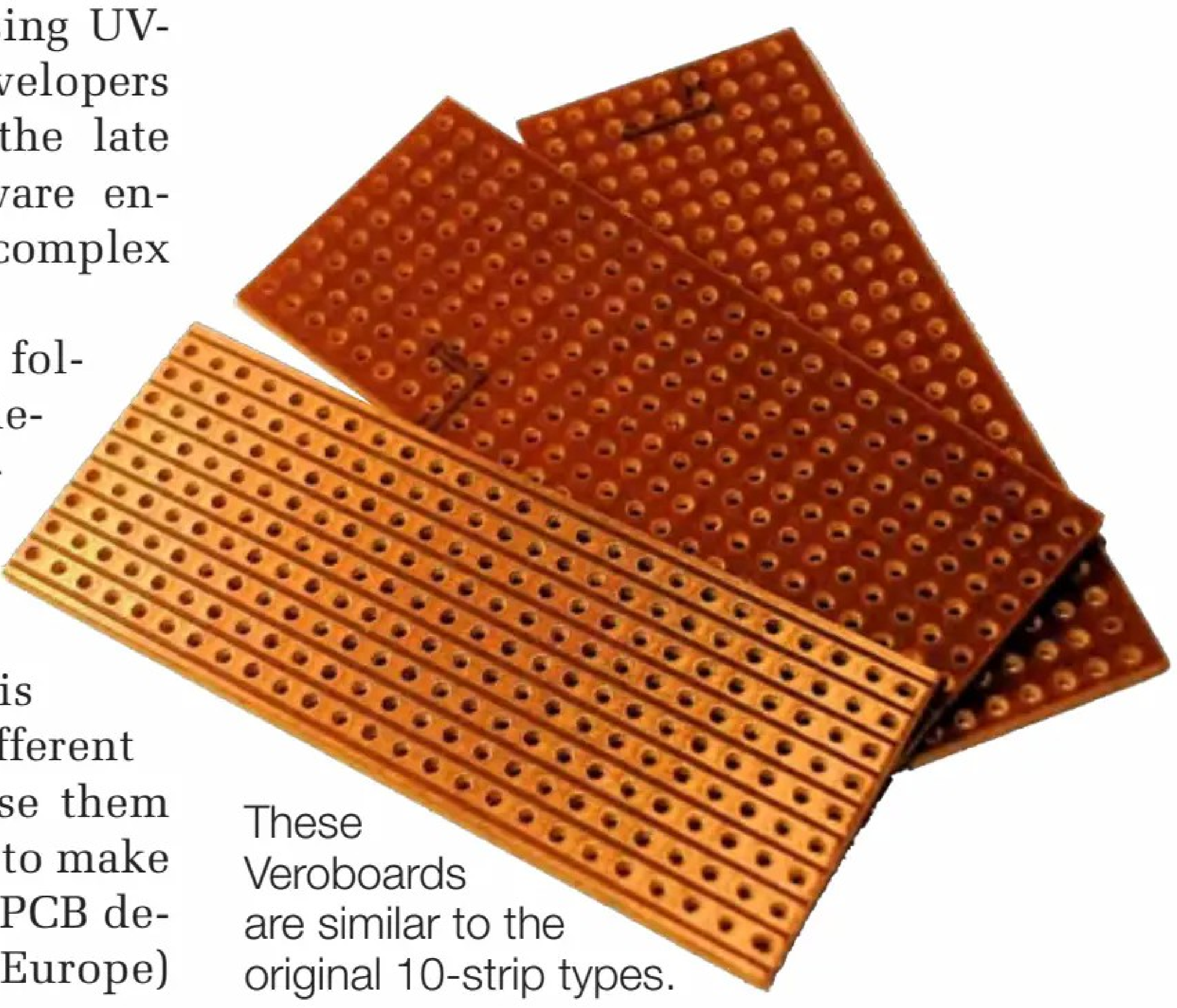
Not to be left out of things, Abacom's LochMaster 4.0 from Germany (*Loch* meaning hole) is Windows software that handles the design of stripboard layouts (Windows 7+, price €49, download from **https://www.electronic-software-shop.com/lng/en/electronic-software/lochmaster-40.html**). Another software product, Ambyr's Stripboard Magic, caught the tide at one time but was discontinued suddenly. It crops up online as 'abandonware'.

They say that the more things change, the more they stay the same. 65 years after its invention, Veroboard is still a staple for building interface circuitry for Arduino or Raspberry Pi projects. Pleasingly, many coders and makers have 'discovered' Veroboard for the first time, and are realising what a wonderfully enabling and adaptable product it really is. It remains ideal for developing smaller discrete circuits, just like early constructors did some sixty years ago.

You can explore the Vero Technologies website at **https://verotl.com/** but note that online ordering is aimed at trade and industry customers. Vero also specialises in machining electronics enclosures to order, perhaps remaining true to its precision machine-tool roots.

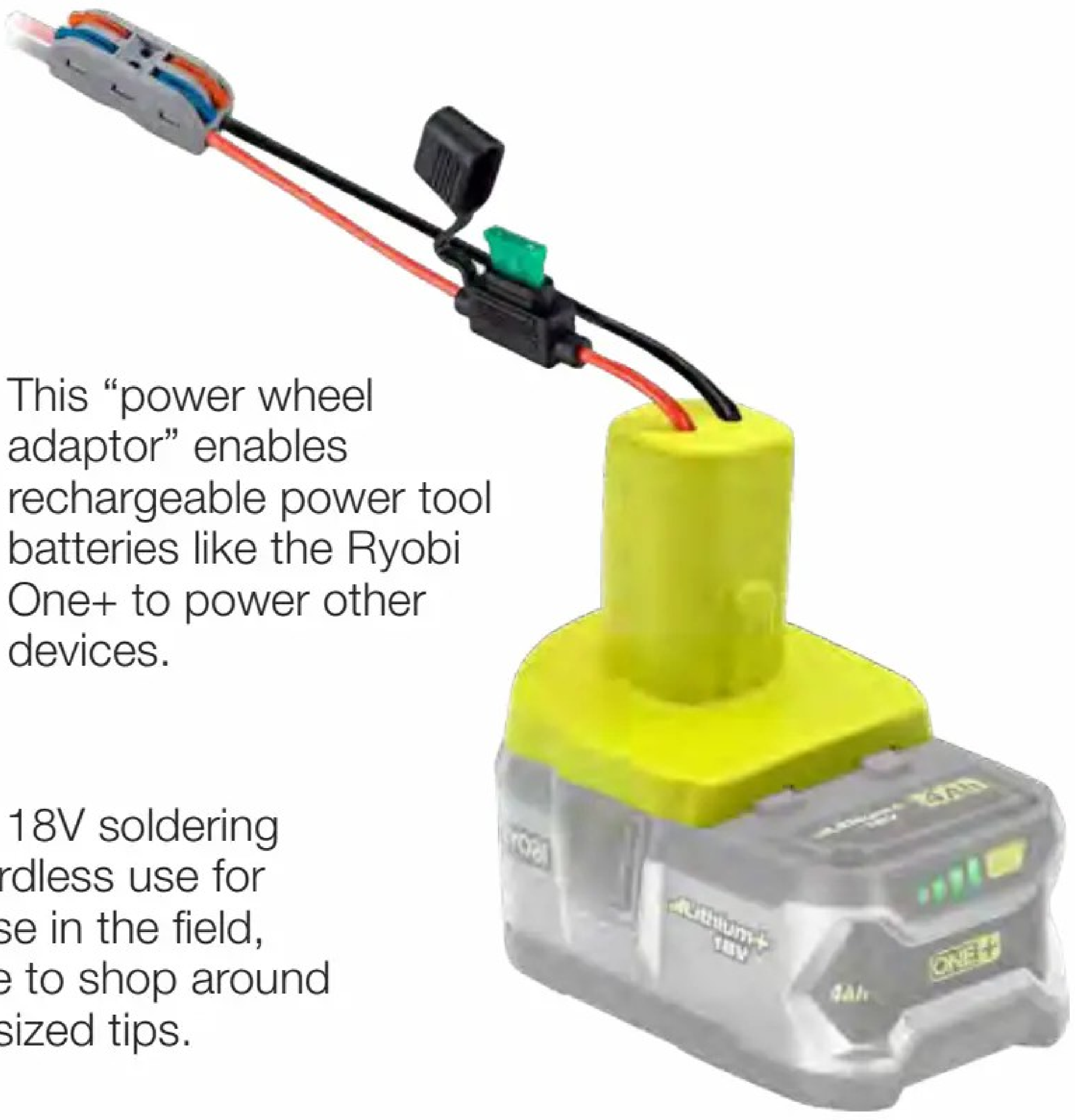
Some hot tips for Ryobi

Back in the February and March 2022 columns, I looked at some rechargeable 18V workshop tools marketed by Ryobi, a brand that's positioned as 'middle-of-the-road' in terms of durability and affordability. There are now several hundred Ryobi One+ power tools and accessories aimed at homeowners, hobbyists, gardeners and DIY enthusiasts alike. They are hugely popular in the USA; a smaller range is sold in Britain.



These Veroboards are similar to the original 10-strip types.

Ryobi's One+ 18V soldering iron allows cordless use for odd jobs or use in the field, but you'll have to shop around for differently sized tips.



This "power wheel adaptor" enables rechargeable power tool batteries like the Ryobi One+ to power other devices.

In my 20 years of experience, some Ryobi accessories have been a lot more successful than others, but I use their One+ 18V soldering iron for occasional quick jobs, or for tasks located out of reach of a mains soldering iron.

Obtaining different-shaped tips has been a problem, though. I prefer using a 3mm bevel tip rather than the chisel or pointed ones bundled with the Ryobi, and replacements from Ryobi cost nearly £18 (see **https://services.ryobitools.eu/uk-uk/getMaterial?matident=5133004382&serident=4000462879**). It appears no other parts, including the element, are replaceable.

Happily, other users reported that tips from the Hakko T18 series also fit the Ryobi (for technical data, see the trade-only website **https://hakko.co.uk/product-category/catalogue/soldering-iron-tips/t18-series/**) and I found compatible T18 types are widely available on AliExpress.

Doubtless most of them are low-cost clones, but they're worth a try, so I bought a selection kit of ten different tip styles. They seem a perfect fit, but the original Ryobi sleeve collar must be used, and great care is needed not to break the exposed element either, or the iron will probably be a write-off.

Loosely connected

With rechargeable 18V batteries in mind, third-party adaptors are sold online that enable popular 18V and 20V ones to fit other brands of power tools. There are also wire-ended types, commonly called "power wheel adaptors" (or with the American spelling, "adapters") that allow them to be used in applications that require a long-life rechargeable power source. Adaptors for DeWalt, Makita, Ryobi and other batteries are listed on the usual websites.

In my garden, I'm using a 'trailcam' (a Victure PIR-triggered digital camera) that records videos of wildlife passing by. I hit on the idea of powering it from an 18V Ryobi battery instead, using a power wheel adaptor hooked up to a regulated power supply, so a likely looking adaptor arrived from China a week or two later.

The popular design shown has an on-off rocker but, like many such imports, it was unbranded and therefore untraceable. A mini-blade fuse protects against overloads – you'll want to derate the 30A one supplied – and a spring-lever connector hooks onto the wire ends.

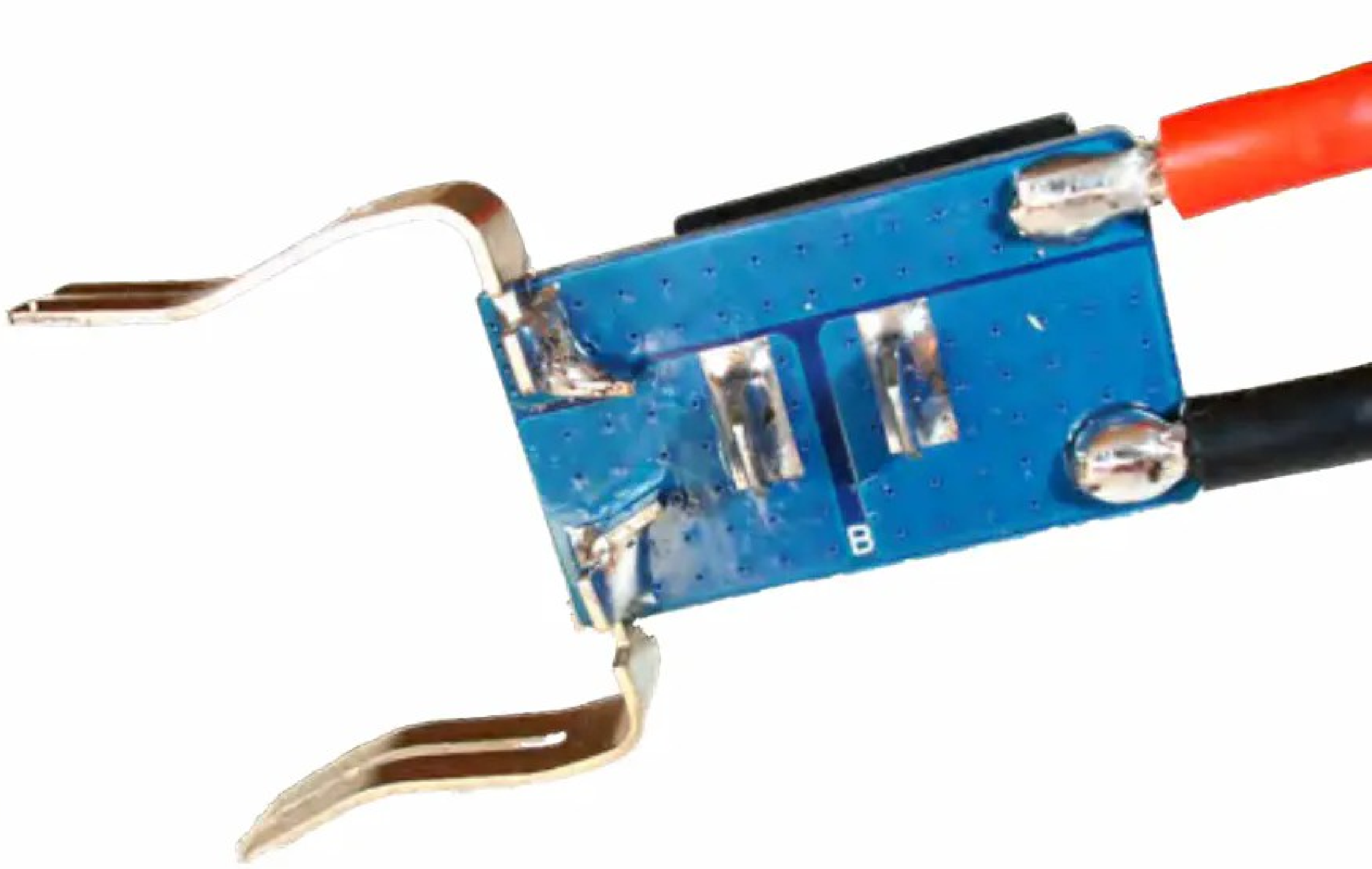
I couldn't resist inspecting the circuit inside, only to find that the battery's +18V wiper contact was soldered perilously closely to the PCB's negative plane. In my sample, there was negligible separation between them. Eventually, through wear and tear, it's not inconceivable that the contacts could wobble or loosen, insulated only by the solder resist, and the battery could risk being shorted out.

It was a poor effort at isolating the battery terminals safely, so I'll use my Dremel to remove the excess metal before going any further.

A suitable DC supply powered from a rechargeable 18V battery could be handy for driving LED lighting, Christmas decorations and novelties outdoors, alarms or monitoring systems in areas where minimum maintenance is desirable or there's no other power available. The battery's high capacity means they should last a decent time, and it's then easy to swap a discharged battery for a fresh one.

That's all for this month. I'll be catching up on the latest news and more besides, in next month's Net Work. You can email the author at **alan@epemag.net** PE

These compatible Hakko-style T18 tips from AliExpress measure identically. The threaded collar shown didn't fit, though.



Inside the author's 'power wheel adaptor', the +18V wiper was perilously close to the PCB's negative plane. It will need to be trimmed down.

MAX'S COOL BEANS

By Max the Magnificent

WEIRD & WONDERFUL ARDUINO PROJECTS

Part 1: DIP LED bar graph displays

Well, hello there. It's so nice to see you. I can't believe we're meeting in the January 2025 issue of Practical Electronics.

I don't want to scare you, but do you realise that we are now a quarter of the way through the 21st Century? I graduated from high school towards the tail-end of the last millennium. That was in the summer of 1975, when the year 2000 was still a quarter of a century into my future.

It feels like I only recently recovered from my enthusiastic contributions to the Y2K celebrations, yet I now find them to be well in the rearview mirror of my life, rapidly receding at a pace that appears to be irresponsibly increasing. I'm too young for all this excitement!

Originally aimed at the experimenter, *PE* began publication in the UK in December 1964. My parents took out a subscription for me starting on my twelfth birthday, in May 1969. At that time, I envisaged the authors of the articles I was eagerly devouring as being an august ensemble of stately scientists, distinguished technologists, and erudite engineers.

I visualised them in deep discourse, capturing their cunning circuits on colossal chalkboards, wearing crisp white lab coats, sporting pocket protectors, wielding slide rules and boasting super-size brains.

It never struck me that I would one day count myself as a member of their guild. I even have my own lab coat, as shown in the photo on my driver's license, reproduced below!

When I became a member of the *PE* community in 1969, the introduction of the first commercial microprocessor, the Intel 4004, was still two years away. Most of the magazine's construction projects in those halcyon days featured discrete (individually packaged) resistors, capacitors, inductors, diodes, and transistors. Some of the projects still employed vacuum tubes.

Now we're going to turn our attention to the shiny new series we have promised above. If you've been a fan of our *Arduino Bootcamp* series, fear not, for towards the end of this article, we shall turn our attention to it one last (?) time.

We will help out a reader who ran into some difficulty driving the 7-segment LED displays (Fig.1) in our example clock circuit. We'll also ponder what could be done to enhance the clock we finished creating last month.

Weird and wonderful

The subtitle to this section does not, in fact, refer to me. Rather, it reflects the fact that, as our new column banner suggests, this article marks the first in a series of *Weird & Wonderful Arduino Projects*.

As usual, my poor old noggin is buzzing with ideas. At some stage, we might even decide to use our existing chronometer as the foundation for our very

They call me "Mad Max", but I've never been able to figure out why...

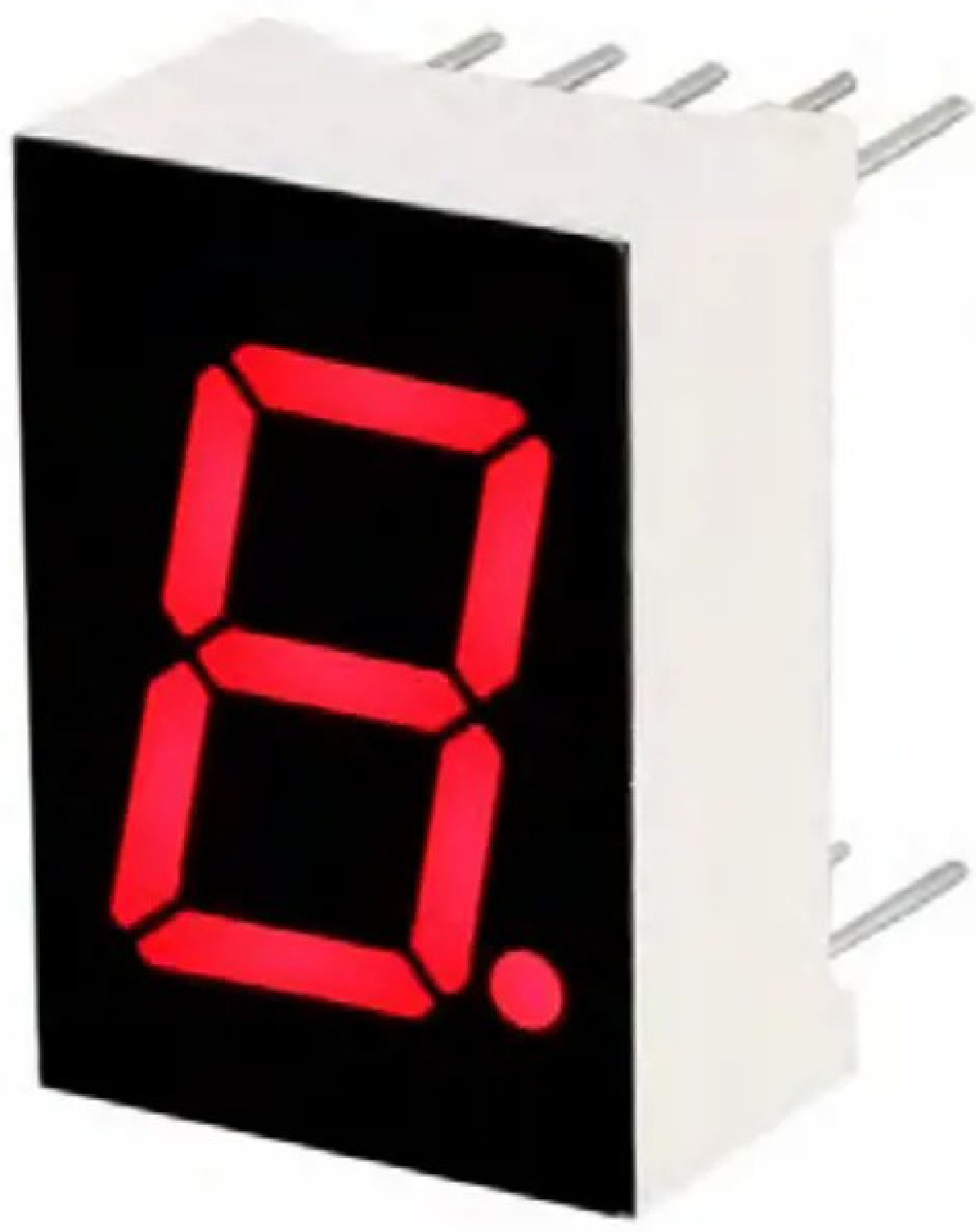


Fig.1: a trusty 7-segment display.

own Vetinari Clock (to be discussed later in this column), for example.

In addition to building some mind-boggling devices, mad scientist style, we will also consider the code required to power them. Along the way, we will explore fundamental hardware and software concepts.

The more we know, the easier things become, and I don't recall ever learning anything that didn't eventually come in handy (sometimes when I least expected it).

And, just for giggles and grins, we are going to peruse and ponder any nuggets of knowledge and tidbits of trivia that tickle my fancy, commencing with…

LED bar graph displays

If you've been implementing the circuits in our *Bootcamp* series, you are already the proud processor of a collection of useful bits and pieces, such as breadboards, jumper wires, resistors, capacitors, LEDs etc. By comparison, if you are starting from scratch, then I've listed the parts we are going to reuse in the "Components" table towards the end of this column.

I should note that the experiments we are going to perform in this column could be implemented using discrete LEDs from your existing 'box of bits'. For myself, however, I really, **really** like LED bar graph displays presented in dual in-line (DIL) packages (DIL package can be abbreviated to "DIP").

These little rascals come in many versions and colours, such as 5-segment displays (**https://pemag.au/link/ac28**), 10-segment displays (**https://pemag.au/link/ac29**), three-colour 10-segment displays (**https://pemag.au/link/ac2a**), and four-colour 10-segment displays (**https://pemag.au/link/ac2b**), to name but a few.

We are going to use yet another variation, a single 8-segment red LED display (Fig.2), but it's always a good idea to get one or more spares. I found a five-pack of these little scamps on Amazon for £7.66 (**https://pemag.au/link/ac2c**), but I'm sure you can obtain them cheaper elsewhere (eg, try AliExpress).

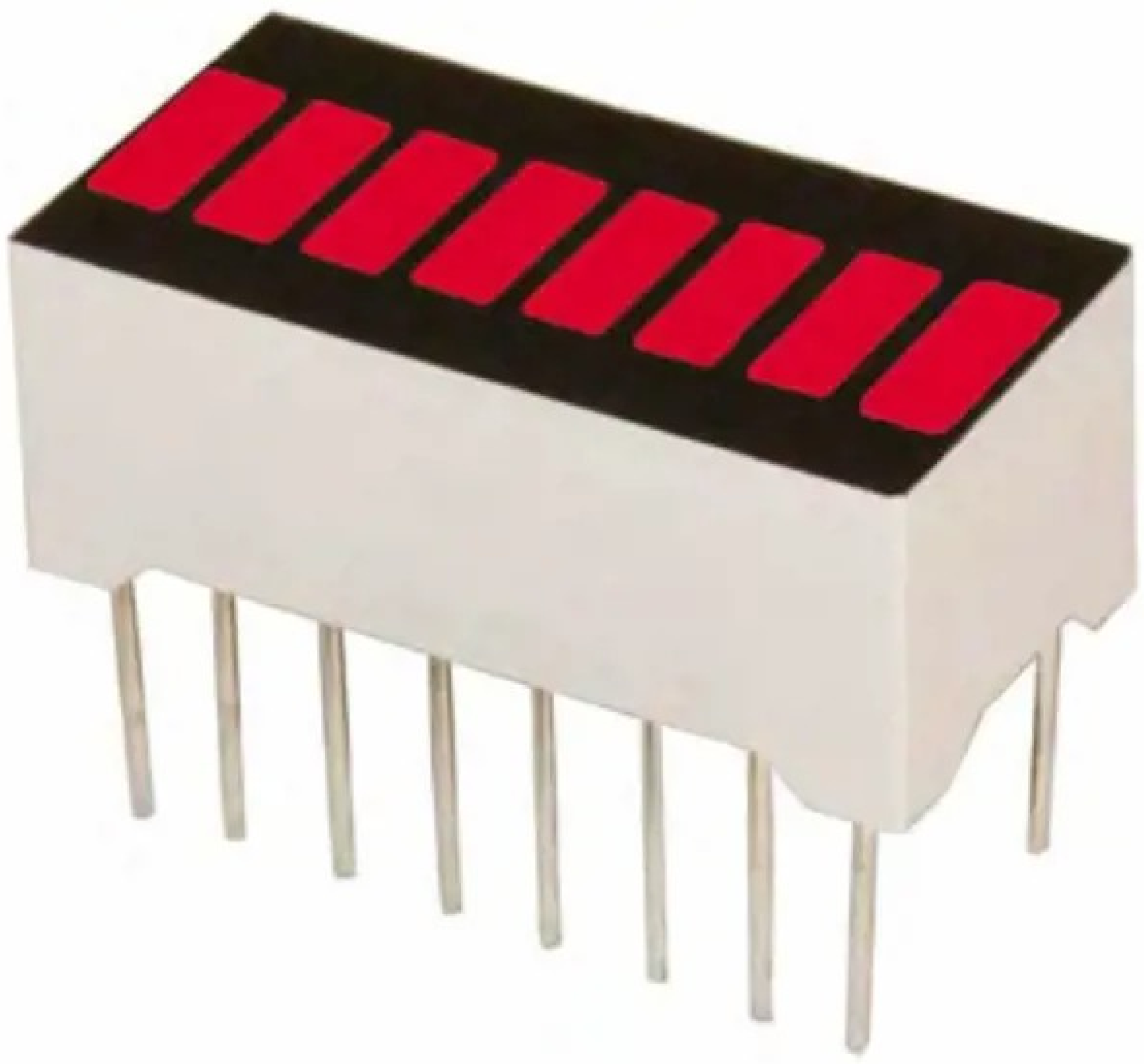


Fig.2: an 8-segment red LED bar graph.

Setting up our breadboard

For the purposes of these experiments, we are going to employ a single full-size breadboard. We will set this up in the usual way (Fig.3).

Any newcomers to our party who are unfamiliar with breadboards can peruse a handy-dandy column on the subject "wot I wrote earlier" (**https://pemag.au/link/ac17**).

There's a lot going on in this diagram, so let's take things step-by-step, starting with the fact that I like to orientate my breadboards such that I have a red power rail at the top and a blue ground rail at the bottom. In this case, we're bringing the power (5V) and ground (0V) from our Arduino Uno into the lower left-hand corner of the breadboard.

The first thing we do is use jumper wires to connect the lower pair of power and ground rails to the upper pair. Also, we insert a 16V 100µF electrolytic capacitor straddling the power and ground rails, as close as possible to where the wires from the Arduino enter the breadboard.

As we discussed in an earlier column (*PE*, August 2024), this type of capacitor looks like a small drink can with two legs. It's a polarised component, which means there are two ways around you could connect it, but only one is correct.

The longer lead is the 'anode', which must connect to the more positive rail (the 5V rail, in our case). The shorter lead is the 'cathode', which is connected to the more negative rail (the 0V rail, in our case). The cathode side of the can is also marked with a stripe with minus signs, to give us a clue.

This capacitor filters out any electrical noise from the power source, which is the Arduino in our case. It will also help to smooth out any power dips caused by components switching and briefly drawing extra current.

With some breadboards, the power and ground rails span the length of the board. Other boards have split rails, in which the rails on the left- and right-hand sides of the board are separated. Split rail boards are useful if we wish to employ multiple power supplies (eg, 5V and 3.3V) and/or multiple ground rails (eg, analog and digital grounds). However, they can be a pain in the rear if you don't realise you are working with this type of board.

If you aren't careful, you can spend an inordinate amount of time trying to debug a seemingly faulty project, only to discover that one side of your circuit is without power! Even worse, the unpowered part of the board can 'appear' to have at least partial power for a variety of reasons, such as power flowing into chips via their signal inputs.

I learned this at my cost, which is why I always add wires linking both sides of the power and ground rails in the centre of the board 'just in case'.

Another thing I do is add a long-tailed header pin somewhere on the lower ground rail. I've shown this on the right-hand side of the rail in Fig.3, but I typically locate it as close as possible to the ground wire coming from the Arduino.

This pin often proves useful later for probing things with our multimeter (it doesn't hurt to have a few more distributed around the ground rails).

Last, but not least, I like to add LEDs at the opposite ends of the power and ground rails to where the power is coming into the board. Whenever I power my board up in the future, these

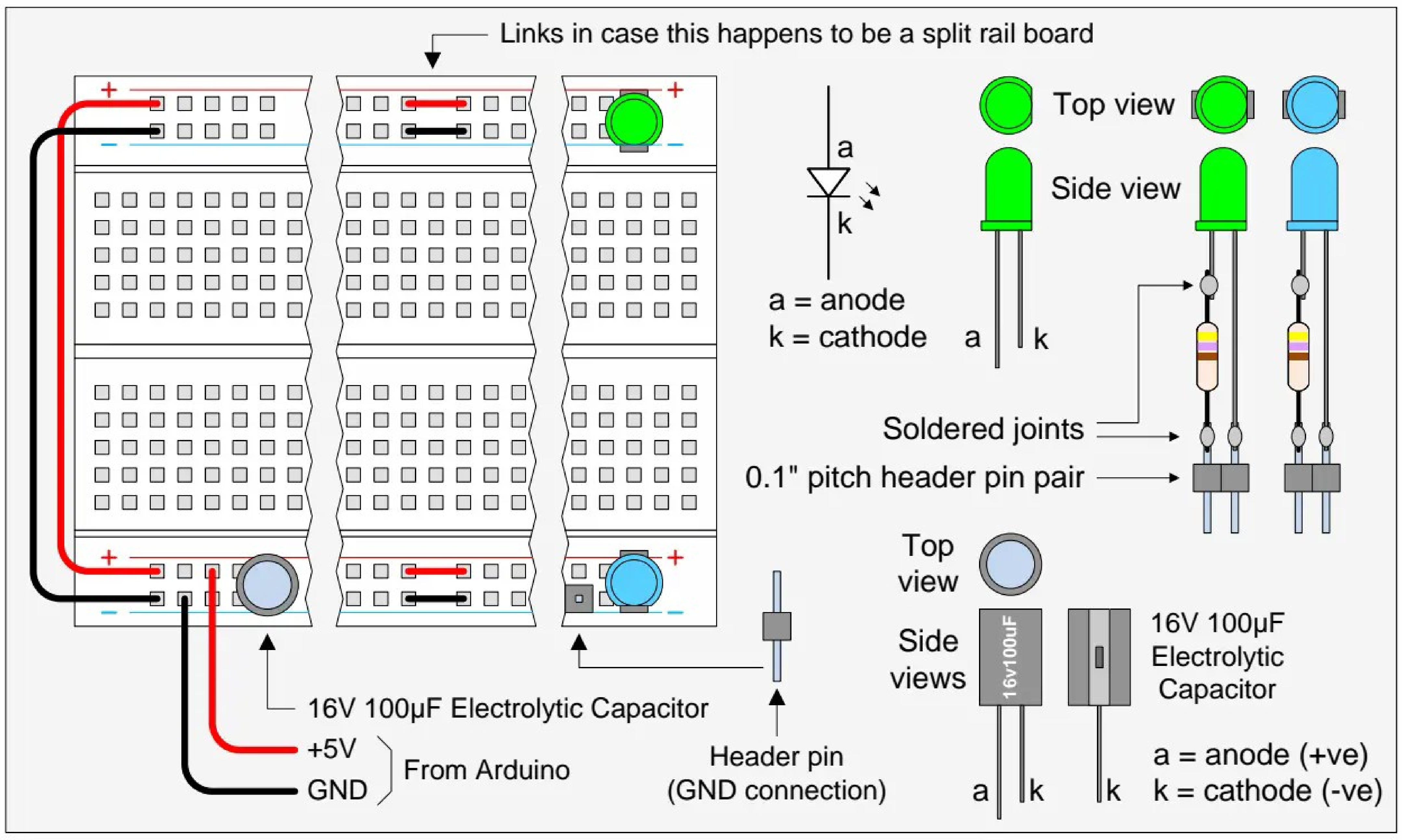

Links in case this happens to be a split rail board
+
–
a
k
a = anode
k = cathode
Top view
Side view
a
k
Soldered joints
0.1" pitch header pin pair
Top view
Side views
16v100µF
a
k
k
16V 100µF Electrolytic Capacitor
a = anode (+ve)
k = cathode (-ve)
16V 100µF Electrolytic Capacitor
+5V
GND
From Arduino
Header pin (GND connection)

Fig.3: our initial breadboard setup.

LEDs immediately inform me that all is right with the world (at least, my breadboard world from a power and ground perspective).

I'm showing green and blue LEDs here. I'm also showing current-limiting resistors of 470Ω (with yellow-violet-brown colour bands).

I'm assuming that these green and blue LEDs have a forward voltage drop of 3V (we discussed the concept of a diode's forward voltage drop in *PE*, March 2024).

Our Arduino Uno gives us a 5V supply, so using Ohm's law of V = I × R, and rearranging things to give I = V / R, we know that I = (5V – 3V) / 470Ω = ~4mA, which will make our LEDs bright enough to be seen without making our eyes water.

We could connect these LEDs and their current-limiting resistors directly into the breadboard. However, as we discussed in a previous column, and as illustrated in Fig.3, it's a good idea to build a bunch of little LED-resistor assemblies, each mounted on a pair of 0.1-inch (2.54mm) pitch long-tailed header pins, that can be quickly and easily reused in future projects.

It doesn't matter whether the resistor is soldered to the anode or cathode terminal of the LED; they just need to be in series with each other. Having said that, I always connect the resistor in-line with the anode terminal because it makes it easy for me to remember which way to plug them into my breadboard.

As we discussed in a previous column, and based on what is to come, if you aren't used to soldering, now is a good time to start learning. A great resource is *The Basic Soldering Guide Handbook*, which was written by *PE*'s very own Alan Winstanley and is available on Amazon (**https://pemag.au/link/ac2d**).

To aid in setting up our breadboard, you can download a PDF of our new setup (with a file name of **CB-Jan25-brd-01.pdf**). As usual, all the files mentioned in this column are available from the January 2025 page of the *PE* website: **https://pemag.au/link/ac2e**

Testing, testing…

If you don't already own a digital multimeter, now would be a really good time to invest in one (we discussed different types, tradeoffs, and how to use these in *PE*, August 2024).

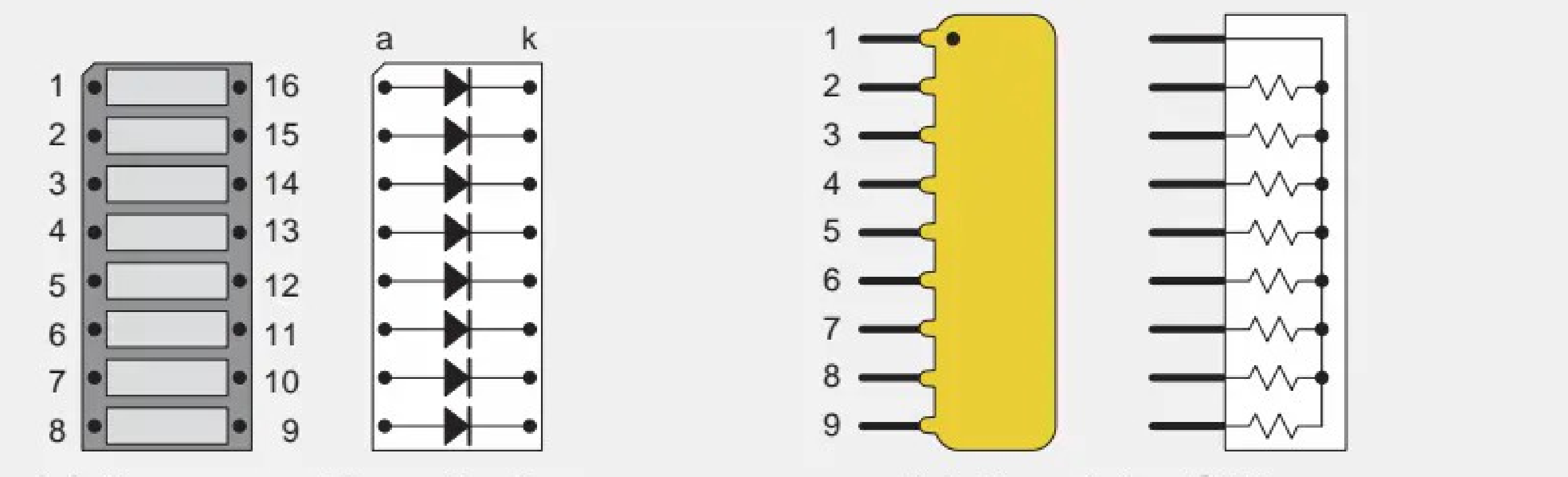
For what we are doing here, I suggest a cheap-and-cheerful manual-ranging device, such as the ULTRICS Digital Multimeter, which is available for only £7.99 from Amazon at the time of writing (**https://pemag.au/link/ac2f**).

Let's start by verifying the probe connections at the multimeter end. The black wire should be plugged into the "COM" port and the red wire should be plugged into the "VΩmA" port.

Plug the USB cable into your Arduino to power everything up and verify that the green and blue LEDs straddling the power and ground rails at the top and bottom of the breadboard light up.

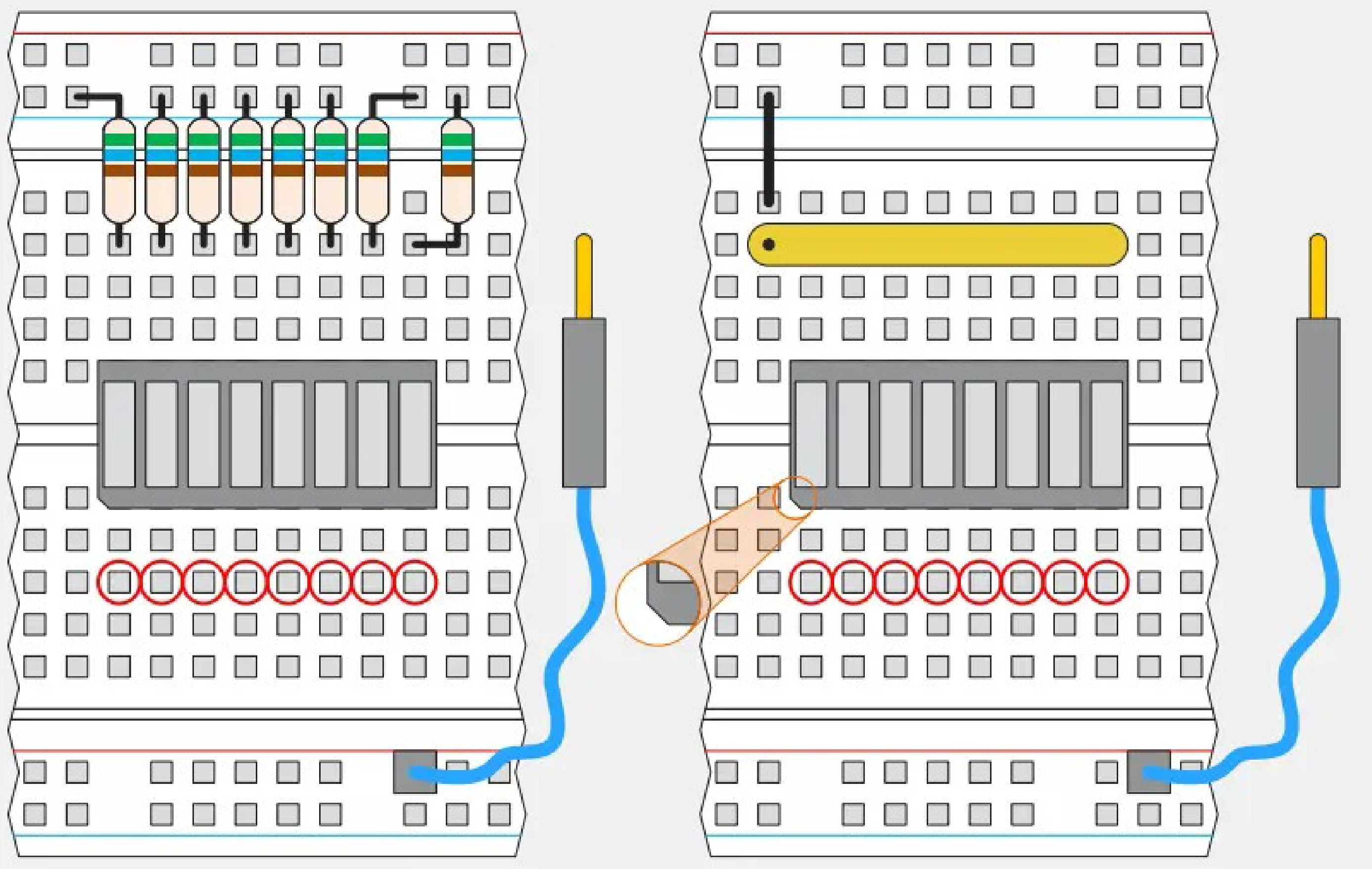
Now, set your multimeter to the 20V DC range and use its probes to check the voltage across the power and ground rails. The easiest (most accessible) points to probe are the header pin connections forming the LED assemblies.

As a rule-of-thumb, I typically apply the black probe first (although the multimeter doesn't 'care' which you connect first). Start with the pair of rails at the bottom of the board and then the pair at the top of the board.



(a) 8-segment bar display

(c) 8-resistor SIP



(b) Using discrete resistors

(d) Using an 8-resistor SIP

Fig.4: adding an 8-segment bar graph display.

Ideally, you should get 5V readings, but the Arduino's supply isn't ferociously robust, so you may have a slightly lower reading, like 4.9V, for example. If your reading is substantially lower than that, you have a problem, possibly even a bad breadboard. If you get a negative reading like –5V, you've probably applied your probes the wrong way around.

Adding our bar graph

Before you do anything else, unplug the USB cable from your Arduino to power everything down. Remember that LEDs can be damaged by electrostatic discharge (ESD), so make sure you are doing everything on your antistatic mat and that you are wearing your antistatic wristband (all this was discussed in *PE*, January 2023).

Observe the 8-segment bar display illustrated in Fig.4(a). As we previously discussed, this is in a DIP.

It can be tricky to determine which is pin 1 on the package. If you look closely, you'll discover that there's a slight chamfer on the pin 1 corner of the plastic. I've shown a zoomed in view of this chamfer in Fig.4(d).

To the best of my knowledge, the LEDs are always presented as shown in Fig.4(a), with their anodes on the left and their cathodes on the right. However, I don't know if this is a rule that's set in stone, and it can be tortuously tricky to track down the data sheets for these components, so I always test things first.

Happily, if you do happen to deploy the device the wrong way around, the only thing that will happen is that your LEDs won't light up.

Plug your bar display and resistors into your breadboard as shown in Fig.4(b). I'm going to use discrete 560Ω resistors (with green-blue-brown bands) as shown.

Why 560Ω? In the future, we are going to use a 74xx595 shift register to drive this display. As we know from our previous column (*PE*, December 2024), the 74xx595's outputs can supply only 6mA. We know that the supply voltage from the Arduino Uno is 5V. We also know that our red segment LEDs have a voltage drop of 2V.

Using Ohm's law of V = IR, and rearranging things to give R = V / I, our ideal resistors would be (5V – 2V) / 0.006A = 500Ω, and 560Ω is the closest (higher) value I have on hand.

And why am I using discrete resistors, instead of the DIL resistor packs we used before? When we look at Fig.4(b), we see that we would have to add a second breadboard to use those devices (we will require the remainder of our current board for future experiments).

There is another alternative: single in-line (SIL) resistor package, where SIL package can be abbreviated to "SIP". These come in myriad sizes and flavours. The one we would use would be a 9-pin package containing eight 560Ω resistors sharing a common pin, as in Fig.4(c).

Using a SIP package would make our lives easier, but I don't have one to hand, which is why I'm using discrete resistors.

Next, add a flying lead as shown in Fig.4(b) and Fig.4(d). Power up your Arduino and use the flying lead to probe each segment anode as indicated by the red circles.

If only one or two LEDs fail to light, you may have a bad display, so try swapping it with your spare. If none light up, the three prime possibilities are: (a) you have a broken flying lead, (b) you've inserted your display the wrong way around, or (c) your display has the anodes and cathodes in the opposite orientation than expected.

Once you have everything up and running and you've verified that all your LEDs are working as expected, power everything down and proceed to the next section.

Connect your Arduino

This is where things start to get even more exciting. Connect digital outputs D2 through D9 from your Arduino to your breadboard as shown in Fig.5 (you can download a PDF of our new setup with a file name of **CB-Jan25-brd-02.pdf**).

Next, let's create a simple program as illustrated in Listing 1 (you can access this code in the file named **CB-jan25-code-01.txt**).

Most of this is self-explanatory. The interesting part occurs in the **for()** loop that starts on Line 36. The index to this loop, **iSeg**, will count from 0 to 7. For each index value, we call the **digitalWrite()** function.

This function has two arguments. The first is the pin to which we wish to write a value. In this case, we are using the current value of **iSeg** to retrieve the required pin number from our **PinsSegs[]** array.

The second argument is the value we wish to write to the selected pin. In this case, we are using Arduino's built-in **random()** function to generate a random value between 0 and 1 (like flipping a coin).

So, why are we using **random(0,2)**? This function generates a random integer between the minimum value (0 in our case) and the maximum value minus 1 (don't ask). Since we've defined our maximum value to be 2, the result will be a random 0 or 1.

The bottom line is that our **for()** loop will assign random 0 and 1 values to the 8 pins driving our display. As we discussed in *PE*, March 2023, the Arduino treats 0 and 1 as being synonymous with **LOW** and **HIGH**, respectively.

Once we've assigned these random values and activated or deactivated the segments on our display, the delay on Line 40 causes it to pause for a second before it does it all again. The result is to present a never-ending series of random bit patterns on the display.

While we're waiting for next month's column, let's all think about some other interesting patterns to display—we can compare ideas when next we meet.

A retro games console

We still have some more experiments to perform before we proceed to our first *Weird and Wonderful* creation. To give you a tiny teasing taste of what is to come, I just tossed a metaphorical penny in the air. Wow! Our virtual coin landed furble side up, which means our first project is going to be a retro games console. I was hoping this would be the case (what are the odds?).

I feel the use of the word "retro" adds a certain something to the proceedings. However, "retro" in this context may be taken to mean "big, bulky, and somewhat ungainly" while, ironically, combining a lack of processing power with a huge hunger for electrical power).

In addition to boasting a 14 × 10 array of tri-coloured LEDs, this will also involve six 7-segment displays

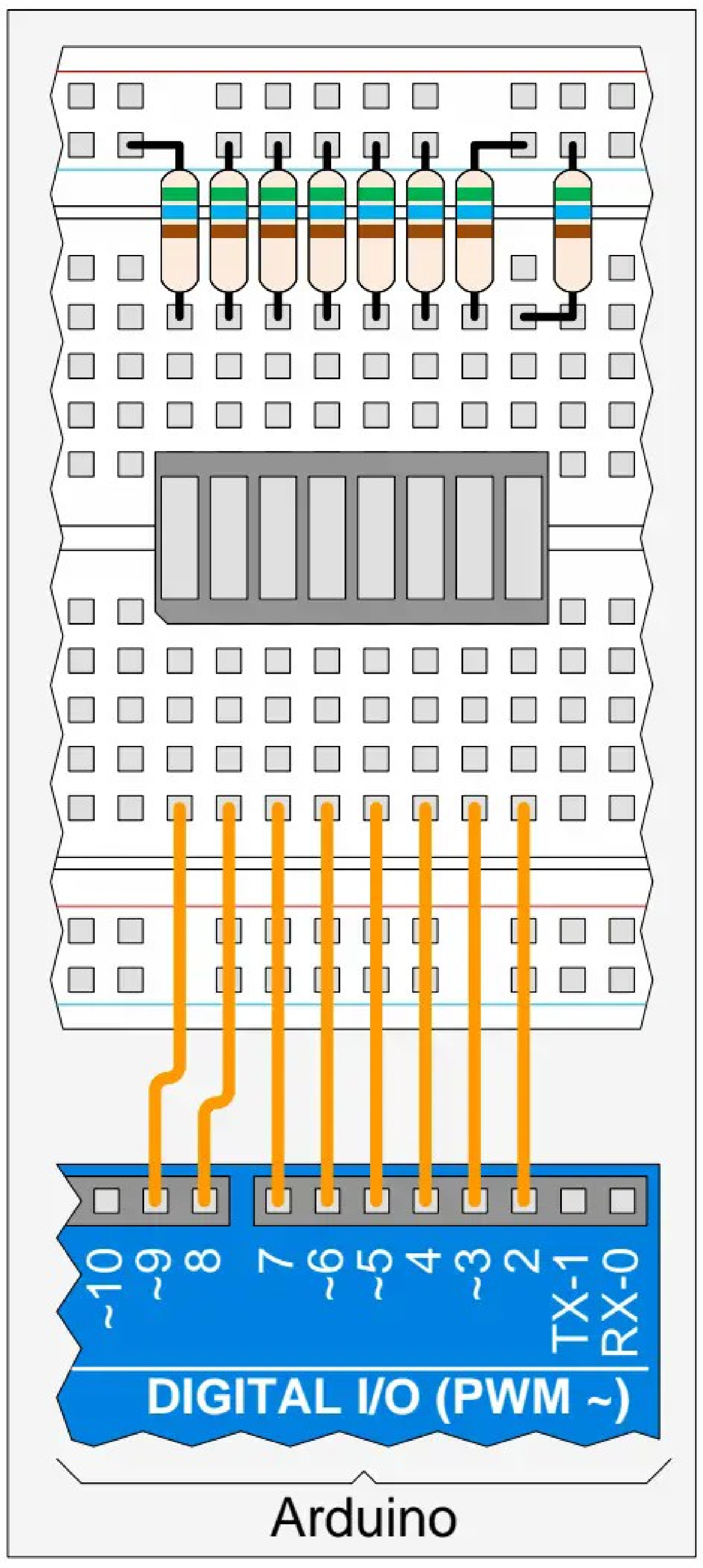


Fig.5: connecting the Arduino.

```
// 8-Segment Displays (General)
#define NUM_SEGS            8
#define SEG_ON           HIGH
#define SEG_OFF           LOW

// Extras
#define CYCLE_TIME       1000

// Declare pins driving the display
int PinsSegs[NUM_SEGS] =
  {2, 3, 4, 5, 6, 7, 8, 9};

// Do this one time
void setup ()
{
  // Initialize everything
  {
     for (int iSeg = 0; iSeg < NUM_SEGS; iSeg++)
    {
      pinMode(PinsSegs[iSeg], OUTPUT);
      digitalWrite(PinsSegs[iSeg], SEG_OFF);
    }
  }

  delay(1000); // Just because
}

// Do this over and over again
void loop ()
{
    for (int iSeg = 0; iSeg < NUM_SEGS; iSeg++)
    {
      digitalWrite(PinsSegs[iSeg], random(0,2));
    }
    delay(CYCLE_TIME);
}
```

Listing 1: our first test program!

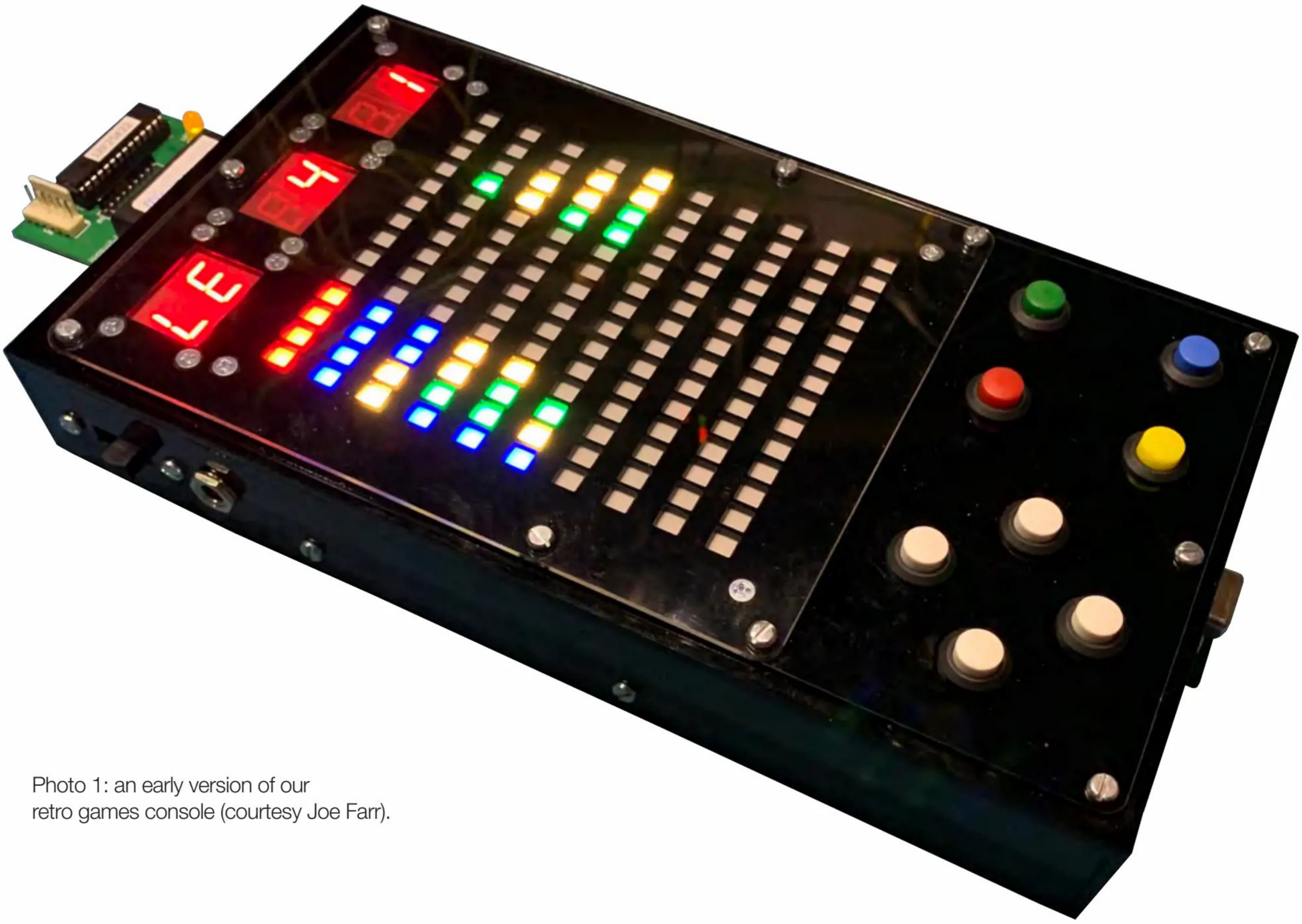


Photo 1: an early version of our retro games console (courtesy Joe Farr).

driven by… wait for it… wait for it… you've guessed it… shift registers! (It's almost as if we have a clue where we are going and what we are doing, but let's not get carried away).

We will, of course, be powering this beast with our trusty Arduino Uno. However, my chum, Joe Farr, has already implemented a PIC-based version of this bodacious beauty (see Photo 1). Apart from anything else, once we've built the base, we can use it as a platform to experiment with various sensors and actuators.

I'm already researching all the bits and pieces we'll need. We'll reconvene in next month's column and commence work on this awesome device that will make you the envy of your family and friends. You will be the first on your street to own one of these soon-to-be collector's items!

Now let's have a look at some of the leftovers relating to last month's final *Arduino Bootcamp* instalment.

Tick, tock, tick…… TOCK!

It probably won't surprise you to learn that I'm a huge fan of the Discworld novels by the late, great Terry Pratchett. One of Terry's characters, Lord Vetinari, is the dictator of the city-state of Ankh-Morpork.

The clock in Lord Vetinari's waiting room maintains accurate time overall, but it sometimes ticks and tocks out of sync: "tick-tock, tick-tock… tick-tock-tick… … tock." In fact, it occasionally misses a tick or a tock altogether.

According to the author, the result is somewhat discombobulating for those for whom an audience is looming.

Why am I waffling on about clocks? Well, if you're a long-standing subscriber to *PE*, you'll know that we've just finished a series of *Arduino Bootcamp* columns.

In those, we used an Arduino Uno R3 microcontroller board and two breadboards to connect various bits and pieces, including a real-time clock (RTC) breakout board (BOB) and four 7-segment displays.

In turn, we used our displays to present the current time in HH:MM format.

Internally, these displays are implemented using light-emitting diodes (LEDs). While LEDs are a relatively recent development, a 7-segment display illuminated by incandescent bulbs was used on a power-plant boiler room signal panel as far back as 1910.

Useful Bits and Pieces from Our Arduino Bootcamp series

Arduino Uno R3 microcontroller module	https://pemag.au/link/ac2g
Solderless breadboard	https://amzn.to/3O2L3e8
8-inch (20cm) jumper wires (male-to-male)	https://amzn.to/3O4hnxk
Long-tailed 0.1-inch (2.54mm) pitch header pins	https://pemag.au/link/ac2h
LEDs (assorted colours)	https://amzn.to/3E7VAQE
Resistors (assorted values)	https://amzn.to/3O4RvBt
Ceramic capacitors (assorted values)	https://pemag.au/link/ac2i
16V 100µF electrolytic capacitors	https://pemag.au/link/ac2j
Momentary pushbutton switches	https://amzn.to/3Tk7Q87
Kit of popular SN74LS00 chips	https://pemag.au/link/ac2k
74HC595 8-bit shift registers	https://pemag.au/link/ac1n

Other stuff

Soldering guide	https://pemag.au/link/ac2d
Basic multimeter	https://pemag.au/link/ac2f

Components for Weird & Wonderful Projects, part 1

4-inch (10cm) jumper wires (optional)	https://pemag.au/link/ac2l
8-segment DIP red LED bar graph displays	https://pemag.au/link/ac2c

Before we proceed, it's worth reminding ourselves that so-called 7-segment displays have an eighth decimal point segment that, as we will discuss, can reflect additional information.

During our *Bootcamp* columns, we experimented with various components. We practised making sounds and playing notes using a passive piezoelectric buzzer. We detected the level of ambient light using a sensor called a light-dependent resistor (LDR), then we used the readings from this sensor to control the brightness of our displays.

We also employed an ultrasonic sensor to ~~annoy our household pets~~ measure distances and present the resulting values on our displays.

Of particular interest were the different techniques we used to drive our displays. We commenced by using eight pins from the Arduino to directly drive the segments on one of our displays. Next, we learned how to multiplex multiple displays.

We then took an in-depth look at binary coded decimal (BCD) and used BCD decoder integrated circuits (ICs) to drive our displays. Finally, we moved to using shift register ICs to control our displays (phew!).

Wait! What?

Literally as I was scribing the preceding section, I received an email from a *PE* community member we'll call Ian (because that's his name).

It seems that Ian had just battled his way through the *PE* November 2024 column in which we transitioned from using 74LS48 (TTL technology) BCD decoders to using CD4511 (CMOS technology) BCD decoders. In his plaintive plea for assistance, Ian spake as follows:

> Hi Max, I need some help. I've had significant success with my clock over the last couple of months, but this latest incarnation is driving me mad. Apart from its other faults, it hates the number 8. Below is an example of what it displays (the dash characters appear as blanks on the displays).

Actual time	Displayed time
14:30	14:30
14:31	14:31
14:32	14:37
14:33	14:37
14:34	14:34
14:35	14:37
14:36	14:37
14:37	14:37
14:38	14:3–
14:39	14:3–
14:40	14:44
14:41	14:41
14:42	14:47

> This sort of pattern repeats itself. Every time it gets to an 8, the display goes blank. It's even worse at 8am when it is supposed to display 8:00 but I get 9:9–. This continues until 8:07 when it displays 9:97, then at 8:10 it displays 8:11.
>
> Every component on the board has been replaced at least once, but it made no difference. I have even completely rewired the board. See the attached photo. I have checked this a million times and am convinced there are no errors. If you have any suggestions as to what I can try, I would be extremely grateful.
>
> Best regards, Ian.

Ian's photo is reproduced below with his permission. Take a few moments to ponder this image. Can you perceive any potential problems?

I did what I usually do at times like this. I FaceTimed my chum Joe Farr

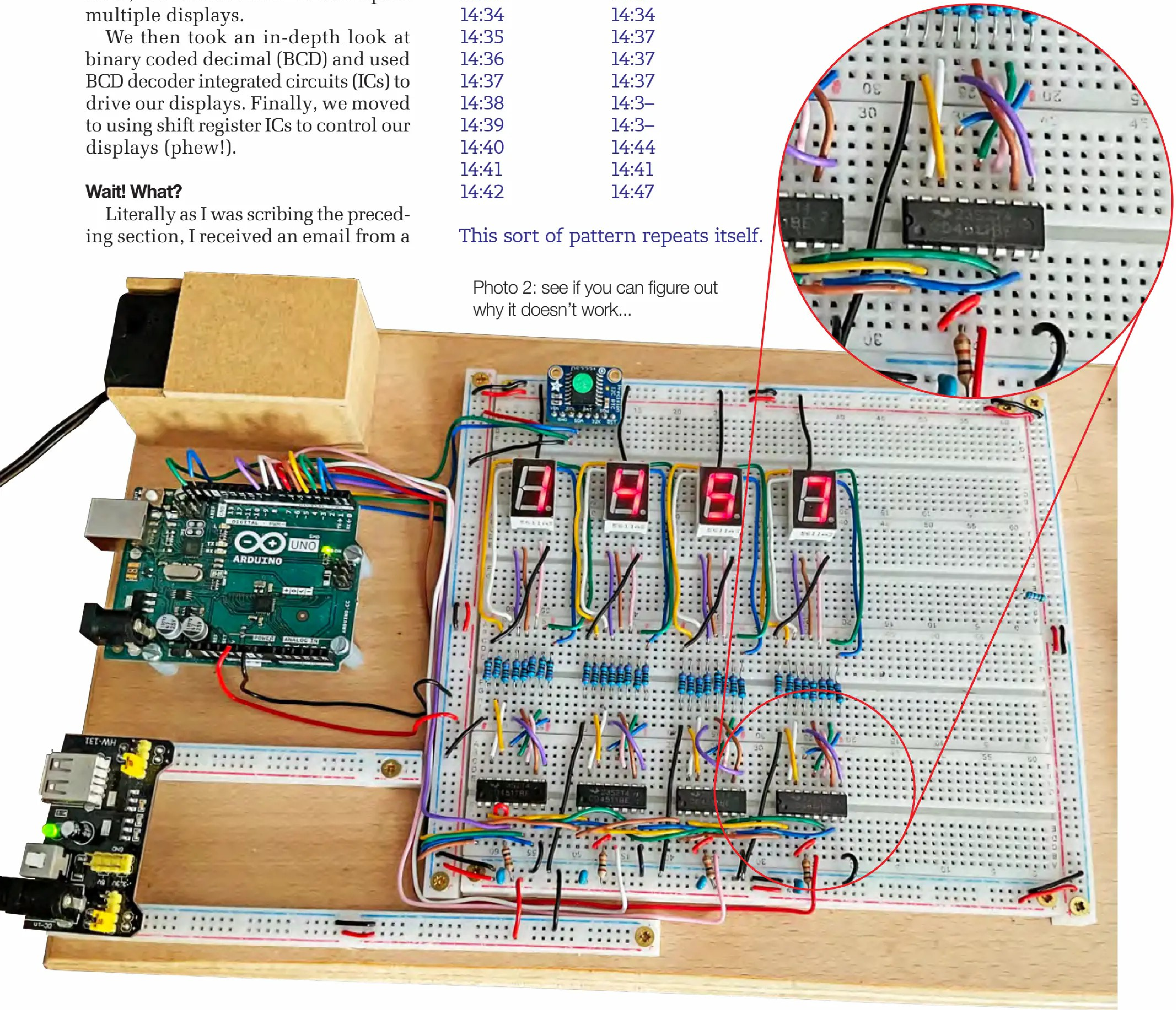


Photo 2: see if you can figure out why it doesn't work...

because (a) debugging someone else's project can be a real pain, (b) "two heads are better than one," as they say, and (c) Joe and I are constantly bouncing ideas back and forth, so this was a good excuse for a chat.

In fact, resolving this one proved to be easy-peasy lemon squeezy (which is rarely the case). When we looked closely at Ian's photo, several problems became apparent:

1. The use of wood screws to secure the breadboards and their power strips to the wooden baseboard is a big "No-No!" These are short circuits waiting to happen. Ian is lucky he didn't short out his power and ground rails.
2. Ian is powering his Arduino via its 5V pin but this is intended to be used as an output, not an input. Supplying power via the 5V pin bypasses the onboard voltage regulator, so any fluctuations or spikes in his power supply could damage the Arduino. Similarly, connecting the USB cable to the Arduino while applying power via the 5V pin also runs the risk of damaging the Arduino.
3. Ian said, "I have checked this a million times and am convinced there are no errors". Well, we've all 'been there and done that'. A closer look at Ian's photo reveals that he forgot to connect the pin 16s (the power pins) on his CD4511s to his 5V supply.

So, how did Ian's circuit manage to work at all? His chips were drawing parasitic power from their data inputs. Most ICs have normally reverse-biased protection diodes from each input to each supply pin. These are intended to provide protection against static shocks (ESD protection) and also against the accidental application of voltage to pins when the chip is not powered.

Because of these diodes, if you apply a voltage to a pin of an IC that's outside its supply range (or any voltage when it is not powered), that voltage can become a power source for the IC. But it won't provide the full voltage or necessarily much current.

This explains why the 8s didn't display because that requires all the segments to be lit and there simply wasn't enough parasitic power to perform the task.

As a related aside, there's an amusing story about the first ARM processor. When engineers at Acorn Computers developed the first ARM (Acorn RISC Machine) processor prototype in 1985, they were surprised to discover that it didn't appear to be drawing any energy from its power supply!

Although they had focused their design on minimising power consumption, they felt the fact it wasn't drawing any power at all was a little extreme. It turned out that, like Ian, they had neglected to connect the power pin, but the chip was drawing just enough power to function from its inputs via tiny leakage currents. However, we digress…

As soon as Ian added those four little wires and properly powered his CD4511s, everything worked as expected and the radiance of his smile brightened the world once more.

All I can say is that I'm scared what Ian thought when he read my column in the December 2024 issue of *PE*, only to discover that I decided to discard our CD4511 BCD decoders and replace them with 74HC595 shift registers!

Endless possibilities

Although our bootcamp clock project is officially over, there are lots of things we could do to enhance its functionality. For example, we could add some pushbuttons and create a menu system.

If so, one of the first tasks we might use our menu system to accomplish is to implement some way to set the current time. Also, we could use it to select between 12- and 24-hour display modes.

We could also add an LDR and use it to implement a nighttime mode, dimming the displays when the ambient light falls below a certain level, such as when the bedroom lights are turned off at night, for example.

Also, we could use our piezoelectric buzzer to add audible accompaniments, such as playing the Westminster Quarters (aka. Westminster Chimes) to mark each quarter-hour.

Furthermore, we could use our menu system to implement an alarm function, setting an alarm time and using our piezoelectric buzzer to play an appropriate wake-up tune.

What about daylight saving time (DST)? We could modify our clock to implement this automatically. Alternatively, we could add a DST On/Off button, or we could employ our menu system to activate/deactivate a DST mode.

Since we ended by using four 8-bit shift registers to control our four 7-segment displays, we now have access to our displays' **dp** (decimal point) segments. In fact, we aren't actually obliged to use the **dp** segments on the displays. We could, instead, use these bits from our shift registers to drive standalone LEDs.

And what could we use these four **dp** segments or standalone LEDs for? Well, we could flash one on and off to mark the passing of the seconds.

We could use one to indicate if the clock was currently in its 12- or 24-hour mode. We could use another to indicate if the clock was displaying regular time or DST. And we could use one to indicate if an alarm has been set.

All the ideas mentioned here are just things I came up with off the top of my head. I'm sure you can think of many more features and functions that, if implemented, would make your cunning chronograph stand proud in the crowd! **PE**

HAMMOND®

1550ZF IP68 flanged die-cast aluminium

Learn more: hammondmfg.com/1550zf

uksales@hammondmfg.com • + 44 1256 812812

Phil Prosser's compact and high-quality

Microphone Preamplifier

If you use microphones for stage, recording or testing, you will be familiar with the need for a preamp to get a usable signal. Many microphones also need 'phantom power'. This small box runs from a plugpack and offers a flat frequency response, very low distortion, low noise and adjustable gain.

Background image: https://unsplash.com/photos/ALM7RNZuDH8

This small microphone preamp is ideal for use in the studio, workshop or on the stage. It allows you to boost the gain of your microphone to line level and delivers a balanced or single-ended signal.

The main version of this Preamp fits into a small, standard-sized enclosure that is widely available, as shown in the photos. This diecast aluminium case makes it tough enough to survive the worst abuse.

If you want to integrate this design into a larger project, we have a version of the board that omits the cutout for the XLR connectors and drops one of the switching regulators, making it easy to run it from existing ±15V DC rails. That would make sense if integrating it into a power amplifier, preamplifier, mixer or similar.

When built as a standalone unit, it runs from 9V DC, as widely used on stage. Those plugpacks generally have 2.1mm plugs with a positive ring and negative tip. We have included reverse polarity protection, so no damage will occur if the wrong plugpack is used.

We set this requirement as it is a fair bet that things will get mixed up on the stage. You don't want to be fiddling with equipment while the crowd waits for the concert to start! Therefore, it should 'just work'.

Performance

The performance of the Microphone Preamplifier depends on various factors. Having low noise is important; the noise level is significantly affected by the source impedance and gain setting.

For a source impedance of 560Ω with 50dB gain and a 1V RMS output, the signal-to-noise ratio is 70dB. At the same output voltage but a gain of 20 times, the SNR is 85dB.

Features & Specifications

- Operates from a 9-15V DC plugpack (9V DC is common for stage equipment)
- Fits in a compact 120 × 93.5 × 35mm diecast enclosure
- Adjustable gain from -15dB to +50dB
- Switchable 20dB attenuator for high-level sources
- Switchable 48V phantom power
- Drives in excess of 5V peak-to-peak (1.75V RMS, 13dBu) into a 600Ω load
- Balanced or single-ended output
- Frequency response: ±0.1dB, 12Hz to 20kHz (gain=26dB/20×) (see Fig.1)
- Signal-to-noise ratio (SNR), Zi = 560Ω, V_{out} = 1V RMS:
  85dB (gain=26dB/20×), 70dB (gain=50dB/320×)
- Total harmonic distortion (THD): <0.002% (see Fig.2)
- Built-in power protection, including reverse polarity
- Input and output protection against most abuse

The frequency response with a gain of 20 times (26dB) is within 0.1dB from 12Hz to 20kHz – see Fig.1.

As shown in Fig.2, the distortion (THD+N) is entirely determined by noise. The underlying distortion is significantly lower, in the region of -95dB (0.0018%) to -105dB (0.0006%).

There is some evidence of noise from the switch-mode regulator at the output, but it is 70-80dB down, depending on the gain setting. That is a low enough level that it is not a concern.

Given that the distortion is so low, it's the SNR that's going to be the performance limit. 70dB is pretty much the worst you can expect as long as your input signal level is sufficient to achieve at least 1V RMS output at the maximum gain setting of around 50dB.

As you reduce the gain to 26dB, it will improve to 85dB, and it should improve further at even lower gain settings, exceeding 90dB. That's assuming your microphone/signal source is high enough in level to still provide a useful output with less gain.

Some challenges

This design is a little tricky because microphone phantom power needs to be 48V DC to be universal. That is a lot higher than 9V DC.

To provide users with headroom of 10-15dB over 0dBu, we want to be able to deliver an output signal with peaks above ±8V. That is needed for people using the mic closer than expected and to deal with loud passages. Stage equipment must have headroom; the sound engineer can deal with levels at the mixing desk.

That means we need supply rails of 48V DC plus dual rails sufficient to get this ±8V from an op amp. We want this in a small box and for the circuit to be as tough as a cheap steak.

If we start with 9V DC and drop 0.5V across a reverse polarity protection diode, then budget another 0.5V for the plugpack output drooping, we only have a poorly-regulated 8V supply to work with. We considered using switched capacitor inverters/doublers using 555s but found that gave marginal supply rail headroom.

After some thought, we decided to take a more industrial strength approach, using two LM2577 boost regulators and a cunning trick to sneak in a negative rail. These regulators are more powerful than we need, but they are widely available and can handle 60V on their output, enough for the phantom power rail.

The resulting power supply fills a significant proportion of the PCB, as we shall see in more detail later. While this solution is hefty, it is very tolerant of input supply variation; even if the output is loaded with a very low impedance, the rails will stay up.

If you are wondering if this could be run from a 9V battery, the answer is not for any length of time. The current draw is far too high to expect a decent lifespan from the battery, and it will go flat exactly when you don't want it to. Full load current draw is about 120mA, which will flatten a 9V battery in short order.

Don't think that all this talk about the power supply means we've forgotten that the preamp part must also have decent performance.

We're using the same hybrid transistor/op amp balanced microphone preamp found in the Loudspeaker Test Jig (published in the June 2024 issue), developed by audio guru Douglas Self. It gives excellent performance with low distortion and noise, plus a wide range of possible gain settings.

Circuit description

Fig.3 is the block diagram for the Preamp, while the main (analog) part of the circuit is shown in Fig.4.

S1 switches phantom power for the microphone via header CON10. Noise is filtered out of the 48V DC supply by a 100Ω/220µF low-pass filter (LPF). We have used 6.8kΩ resistors for the two

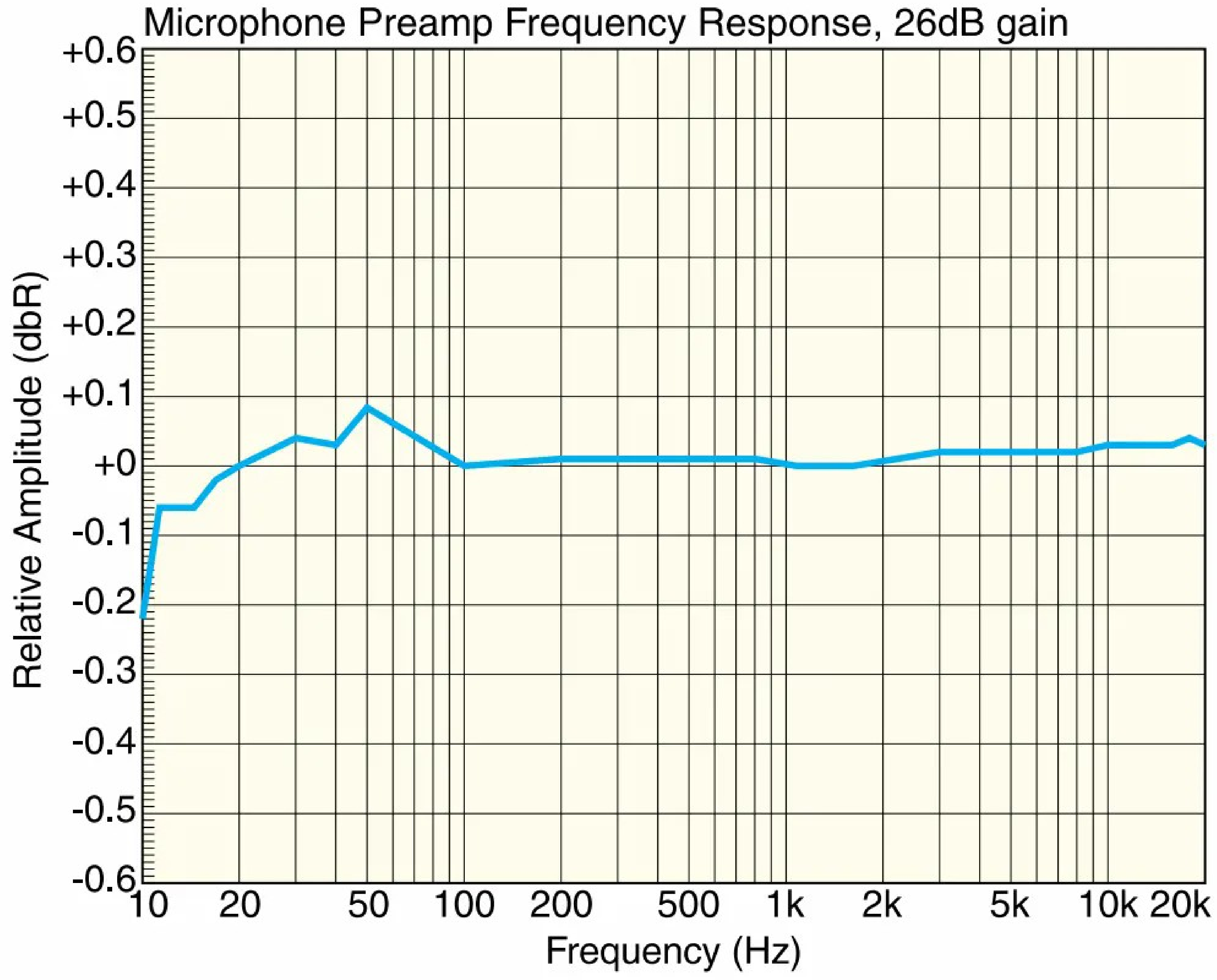


Fig.1: we had to make the vertical scale very small to see the variations in frequency response as it is so flat.

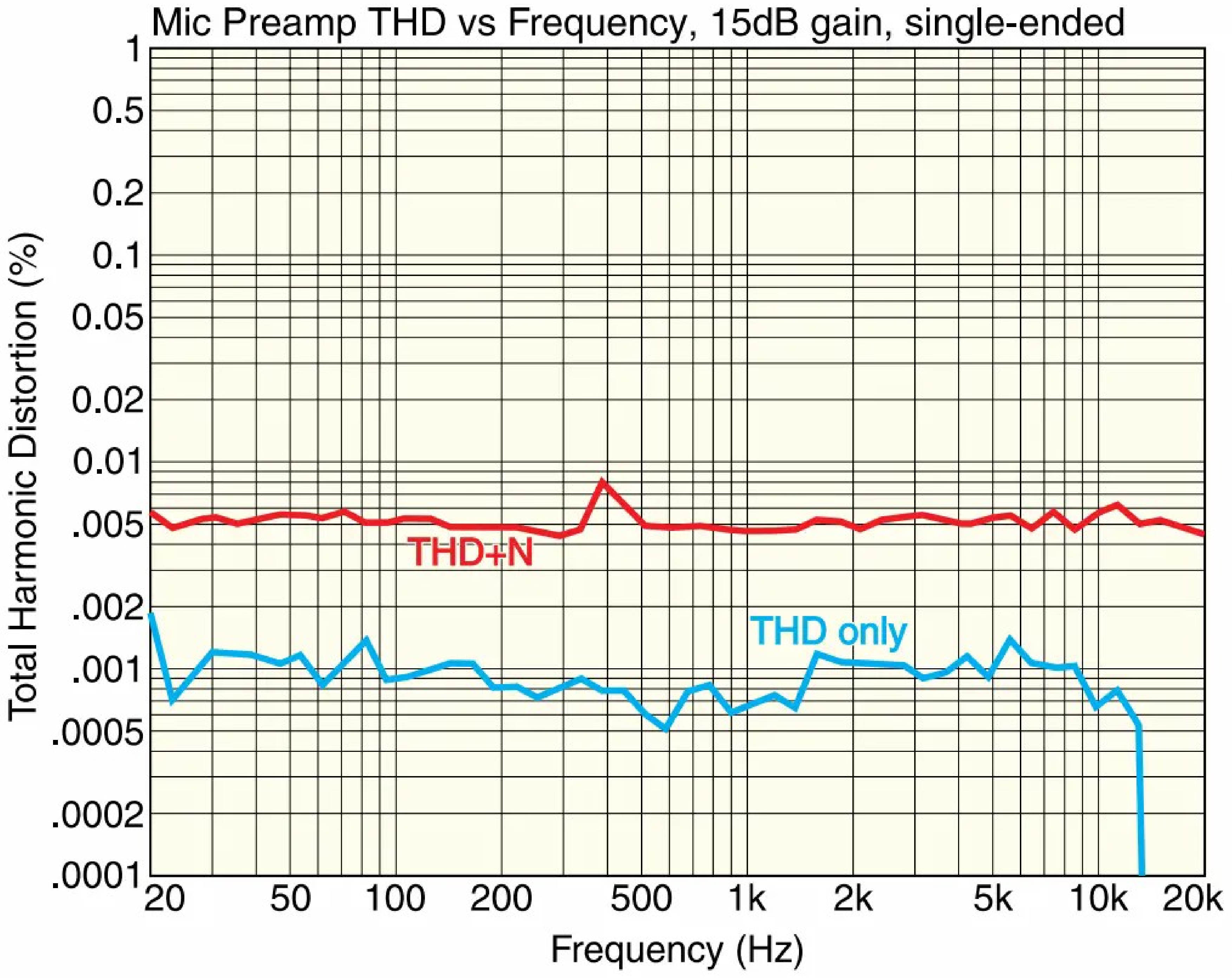


Fig.2: any distortion produced by the circuit is well and truly buried in the noise. Thus, the SNR is the primary determinant of the performance at any given gain setting.

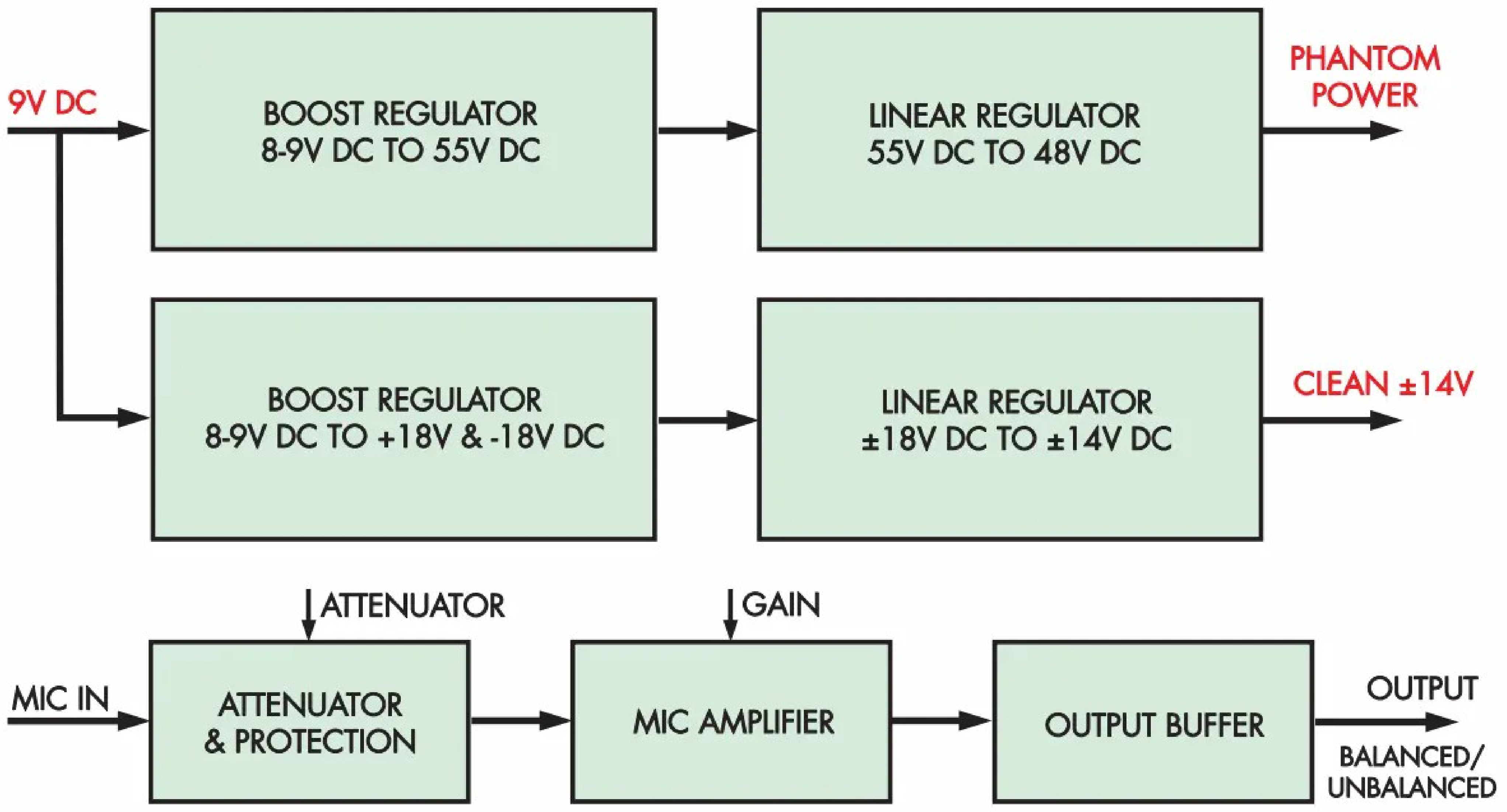


Fig.3: the Mic Preamp block diagram shows the somewhat complicated power supply at the top, with the superficially simple attenuation and preamplification circuitry below.

bias resistors; these should be matched as close as possible.

We selected two resistors that measured within 0.1% from our collection of 6.8kΩ 1% resistors. You could buy 10 resistors and choose the best-matched pair.

The 47μF/100nF parallel capacitor pairs block DC from the microphone signal as it's fed into the attenuator. These prevent the full 48V phantom power from being applied to the attenuator when the microphone is unplugged, so they must be rated at a minimum of 63V.

This Preamp has a 20dB pad at the front end. It can be switched in to avoid the Preamp clipping with higher-level input signals. The pad uses two 1.8kΩ resistors in series with the input signals and a 430Ω resistor connected between the terminals of RLY1.

When the attenuator is switched

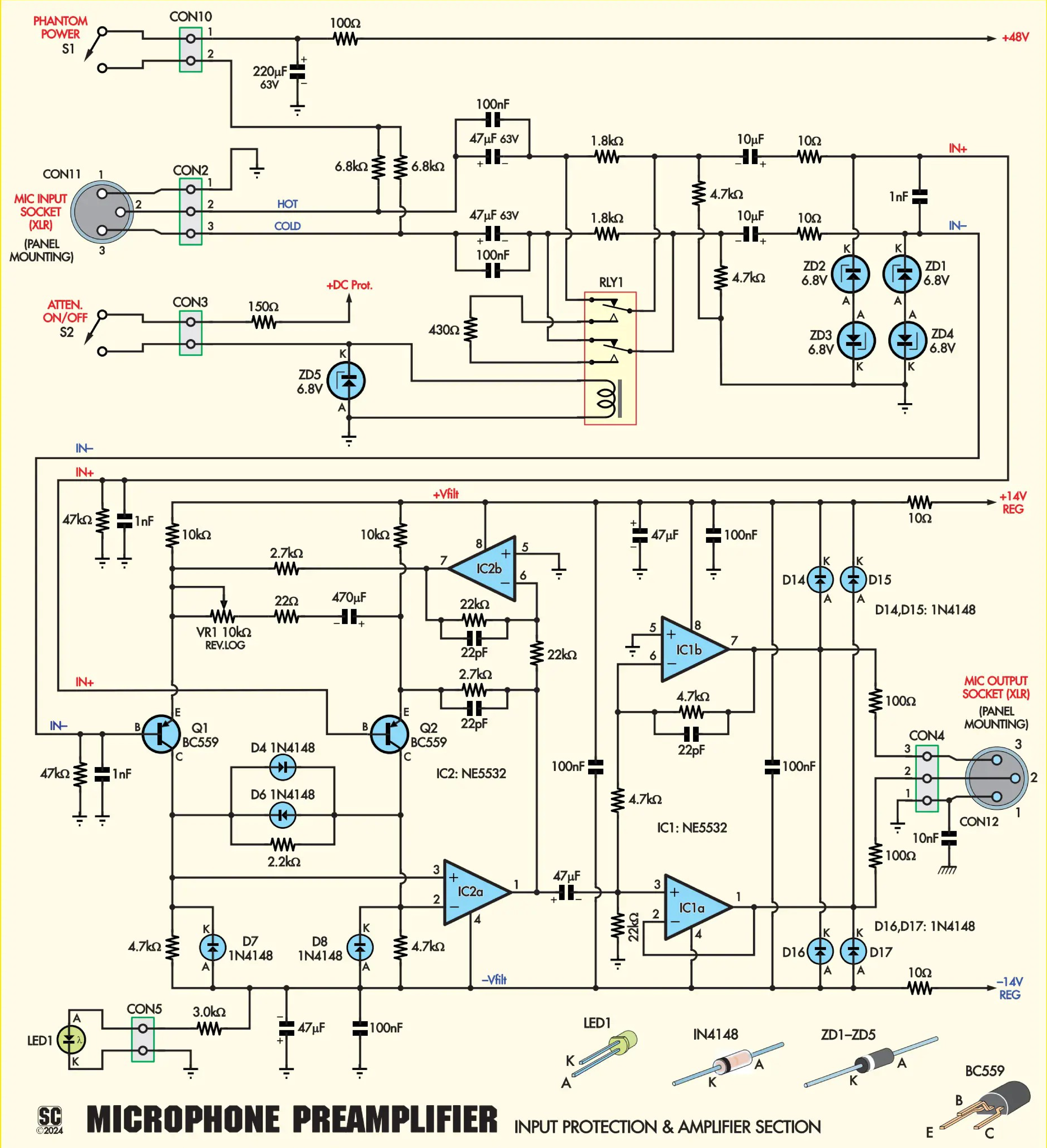

PHANTOM POWER
S1
CON10
100Ω
+48V
220μF 63V
100nF
47μF 63V
6.8kΩ
6.8kΩ
1.8kΩ
10μF
10Ω
IN+
CON11
MIC INPUT SOCKET (XLR)
(PANEL MOUNTING)
CON2
HOT
COLD
4.7kΩ
1nF
IN–
47μF 63V
100nF
1.8kΩ
10μF
10Ω
ZD2 6.8V
ZD1 6.8V
4.7kΩ
RLY1
+DC Prot.
ATTEN. ON/OFF
S2
CON3
150Ω
430Ω
ZD3 6.8V
ZD4 6.8V
ZD5 6.8V
IN–
IN+
+Vfilt
+14V REG
47kΩ
1nF
10kΩ
10kΩ
2.7kΩ
IC2b
47μF
100nF
10Ω
22Ω
470μF
22kΩ
22pF
22kΩ
D14
D15
D14,D15: 1N4148
VR1 10kΩ REV.LOG
IC1b
IN+
2.7kΩ
22pF
4.7kΩ
22pF
100Ω
MIC OUTPUT SOCKET (XLR)
(PANEL MOUNTING)
CON4
IN–
Q1 BC559
Q2 BC559
D4 1N4148
D6 1N4148
2.2kΩ
IC2: NE5532
100nF
100nF
47kΩ
1nF
4.7kΩ
IC1: NE5532
CON12
10nF
100Ω
IC2a
47μF
IC1a
22kΩ
4.7kΩ
D7 1N4148
D8 1N4148
4.7kΩ
D16
D17
D16,D17: 1N4148
–Vfilt
10Ω
–14V REG
LED1
CON5
3.0kΩ
47μF
100nF
LED1
IN4148
ZD1–ZD5
BC559
SC ©2024
MICROPHONE PREAMPLIFIER
INPUT PROTECTION & AMPLIFIER SECTION

Fig.4: the main analog section of the Preamp circuit. It is based on two dual op amps and two transistors; the transistors lower the noise floor substantially. The second op amp drives the balanced and unbalanced outputs. Relay RLY1 switches in a resistive attenuator so it can handle higher level input signals.

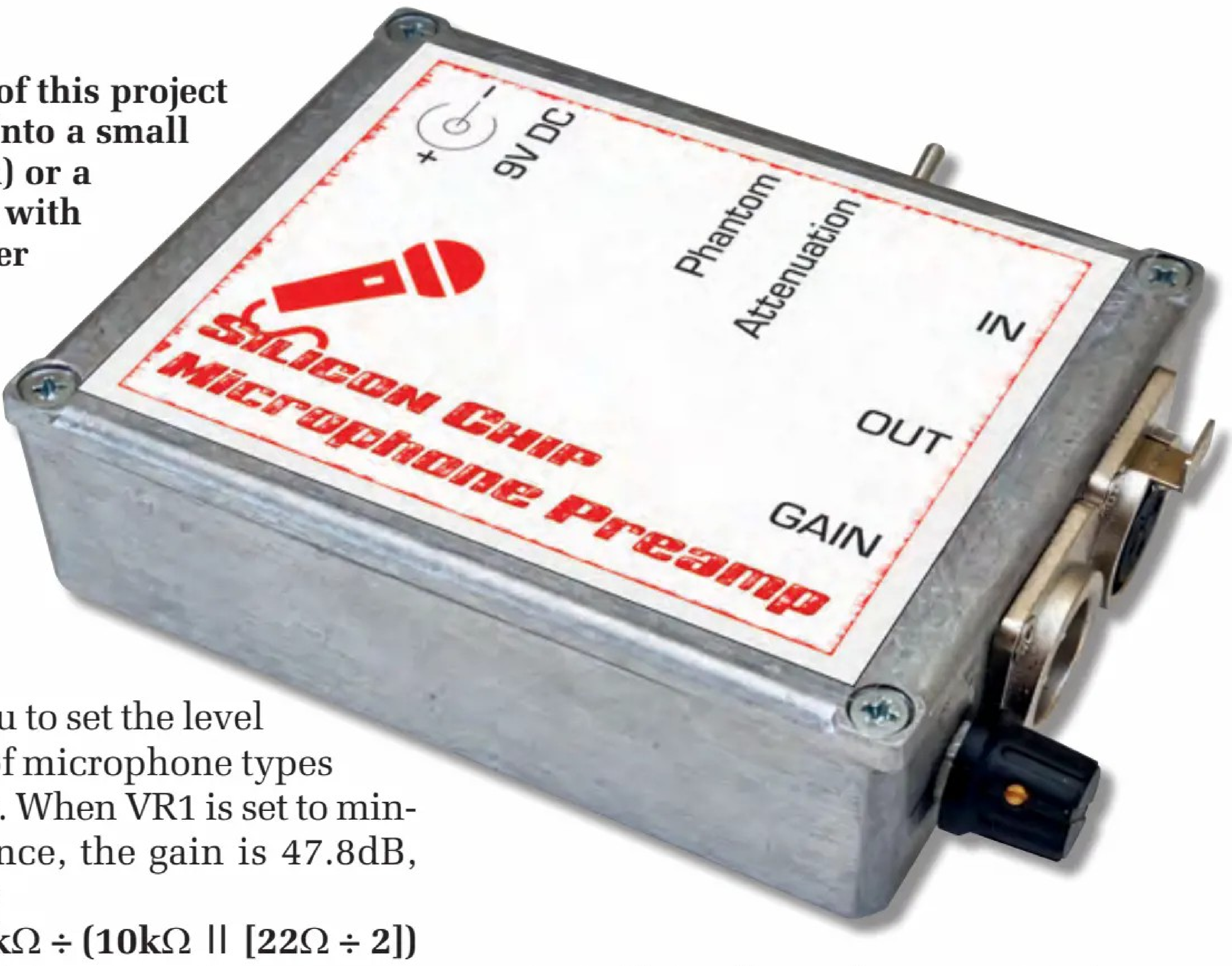
out, the relay shorts out the 1.8kΩ resistors, and the 430Ω resistor is out of the circuit. When switched in, the 430Ω resistor is connected between the downstream ends of the 1.8kΩ resistors, forming a voltage divider. These relatively low values minimise additive noise from the attenuator and keep the impedance driving the following preamplifier low.

To calculate the attenuation of this stage (when activated), add a mental ground connection in the middle of the 430Ω resistor, splitting it into two 215Ω resistors. These resistances are in parallel with the 4.7kΩ resistors to ground, so the dividers are formed with resistances of 1.8kΩ plus the microphone source impedance and 205.6Ω. Assuming a low source impedance, the resulting attenuation is -19.8dB.

Note that we need closely-matched values for the 1.8kΩ and 4.7kΩ parts to ensure good common mode rejection performance when the attenuator is switched in.

We have used a relay for this job as our experience with switching small signals with miniature toggle switches wired to the board is not great. A telecom relay gives better long-term reliability and lower noise for a modest increase in cost.

The 6.8V zener diode across the relay protects it in case someone runs the Preamplifier from a higher voltage than expected. The series resistor will get quite warm, but it should survive, provided this abuse is not continuous.



Two versions of this project allow it to fit into a small box (as shown) or a larger chassis with dual-rail power available.

Preamp gain

We have provided a variable gain that allows you to set the level from a range of microphone types and situations. When VR1 is set to minimum resistance, the gain is 47.8dB, calculated as:

$$G = 1 + 2.7k\Omega \div (10k\Omega \parallel [22\Omega \div 2])$$
$$G = 1 + 2.7k\Omega \div 10.98\Omega$$
$$G = 247\ (47.8dB)$$

When VR1 is set to its maximum of 10kΩ, the gain is 5.1dB:

$$G = 1 + 2.7k\Omega \div (10k\Omega \parallel [(10k\Omega + 22\Omega) \div 2])$$
$$G = 1 + 2.7k\Omega \div 3338\Omega$$
$$G = 1.8\ (5.1dB)$$

Using a reverse log taper potentiometer for VR1 results in the attenuation being ‘linear’ in dB terms as the potentiometer is rotated. Otherwise, most of the potentiometer’s range will result in relatively low gain, with the last fraction of the rotation ramping the gain over 20dB or so. So make sure the pot you choose has a ‘reverse log’ or ‘reverse audio’ (C) taper.

The small signal diodes in the preamplifier (D4-D8) ensure the op amp inputs are not overdriven. We have included a buffer following the preamplifier that also produces an inverted output. This allows the output to be single-ended or drive a balanced line at a high level.

We have also added small signal diodes to the positive and negative rails on the outputs (D14-D17) so that if someone inadvertently connects this to a piece of equipment with a large DC offset on its input, they will protect the NE5532 (IC1).

We have incorporated 100Ω series resistors on the outputs to ensure the op amp remains stable even when driving difficult loads or long cables. Those will also help to limit the current flow in the case of a misconnection. You can use the positive buffered output at pin 2 if you only need a single-ended output.

Power supply

The power supply portion of the circuit is shown in Fig.5. The overall design comprises two switch-mode pre-regulators that drive LM317/337 linear regulators. This generates very clean power rails, including the phantom power rail.

The phantom power supply uses the LM2577 (IC3) in a textbook configuration. Its input is bypassed with a 220µF low-ESR capacitor and a 100nF capacitor. 220µF is quite low, but the maximum current we need to supply is less than 30mA. That is little more than idling for the LM2577.

We have increased the compensation capacitor in series with the 2.7kΩ resistor at its pin 1 from a suggested value of 1µF to 10µF. That slows the startup of the boost regulator. Our small 500mA switchmode plugpack went into current limiting without this; that

Calculating the total current draw

The phantom power supply needs to provide about 10mA to the LM317HV (REG1) and a maximum of 14.1mA into the 6.8kΩ resistors if they are shorted to ground. That is 24mA at 55.3V, which will require ~166mA (55.3V ÷ 8V × 24mA) at the input of the REG3.

The dual rail power supplies must supply up to about 40mA to the NE5532 op amps and input circuit and about 10mA each for REG3 and REG4. That is a total of 100mA, given there are positive and negative rails, meaning a draw of up to about 225mA (18V ÷ 8V × 100mA) at the input to REG4 in the worst case.

That means the Preamp could draw something in the region of 350mA, although that would only happen if it were driving a shorted load. Most 9V plugpacks can supply this, but most 9V batteries can’t. The most we saw in our tests was 150mA from 9V. Note that the worst case current is at startup, when the switch mode regulators are charging the 56V and ±18V supply filtering capacitors.

We have included a power LED, powered from the negative rail. We chose this rail because if a user connects the Preamp to an 18-24V DC plugpack, the boost regulator for the positive rail will likely shut down, and the negative rail will not be generated. No damage should occur, but the user will be informed that it is not operating by the power LED being off.

would not be a problem with a larger plugpack (or a linear type).

The output voltage is set by the resistors connected to the feedback pin (pin 2). With the 33kΩ/750Ω feedback divider and IC3's internal 1.23V reference, the result is an output of 56.25V (1.23V × [33kΩ ÷ 750Ω + 1]). A 10Ω/10μF low-pass RC filter on the output reduces the remnants of the 52kHz switching frequency.

The following LM317HV-based linear regulator drops the output close to the 48V required for phantom power while removing most of the remaining switch-mode noise. The 330Ω and 12kΩ feedback resistors set its output to 46.7V (1.25V × [12kΩ ÷ 330Ω + 1]).

Switch-mode regulator IC4 produces the +18V rail (dropped to +14V by linear regulator REG3) and is set up similarly to REG3. It uses the recommended 1μF compensation capacitor rather than the higher 10μF value used for REG3 to reduce its startup current. A lower value inductor of 100μH is used due to the much lower boost ratio required, under 2:1. You must use toroidal inductors.

Its output voltage is set by 33kΩ and 2.4kΩ resistors to about 18.4V (1.25V × [33kΩ ÷ 2.4kΩ + 1]). It also has a 10Ω/10μF low-pass RC filter on its output, and the following LM317-based linear regulator has its output voltage set by 3.9kΩ and 390Ω resistors, resulting in about 13.75V (1.25V × [3.9kΩ ÷ 390Ω + 1]) for the positive op amp rail.

Now to the cunning trick. Being a boost regulator, LM2577 (IC4) switches its pin 4 to ground to establish a current in L1.

When pin 4 subsequently goes open-circuit, that current continues to flow and charges the output capacitor to our target of 18V DC. That is repeated at 52kHz by this device. Therefore, we have a node at pin 4 switching between about 18.7V and ground.

Our trick is to generate the negative rail is piggybacking off this node using a 2.2Ω resistor, 47μF capacitor and ultrafast diode D9. When the output of IC4 reaches 18.7V, that capacitor is charged to around 18V via D9. When IC4 switches pin 4 to ground, the positive end of that capacitor is pulled to 0V, so the negative end goes to about -18V. That charges the following 47μF capacitor via diode D3, creating our negative rail.

The negative rail is not directly regulated, but the positive rail regulation will ensure the negative rail is about right. LM337 linear regulator REG4 has its output set to -13.75V, so even if its input is a little lower in magnitude than that of REG3, the

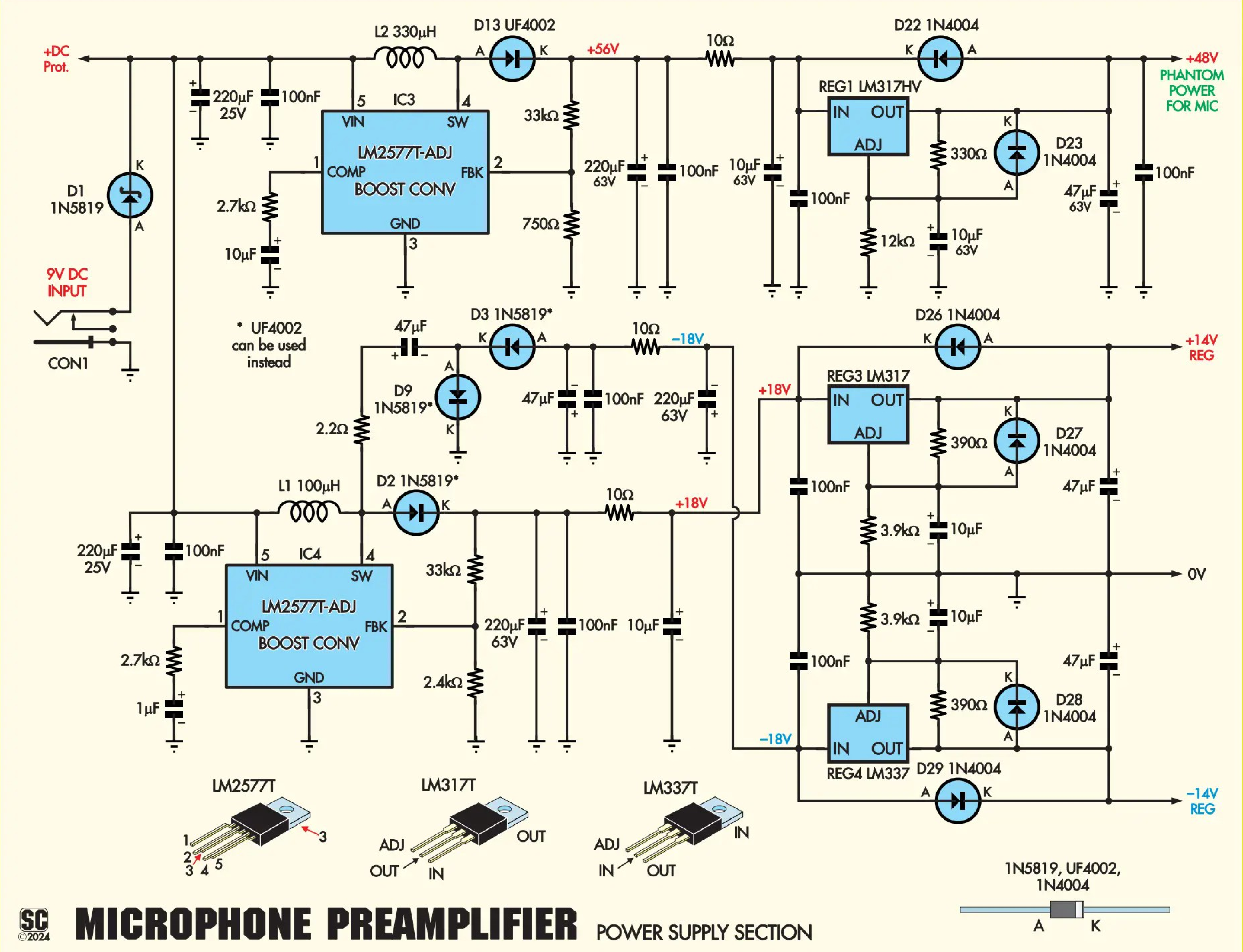

+DC
Prot.
L2 330μH
D13 UF4002
A
K
+56V
10Ω
D22 1N4004
K
A
+48V
PHANTOM
POWER
FOR MIC
220μF
25V
100nF
5
IC3
4
VIN
SW
LM2577T-ADJ
COMP
FBK
BOOST CONV
GND
1
2
3
33kΩ
750Ω
220μF
63V
100nF
10μF
63V
REG1 LM317HV
IN
OUT
ADJ
100nF
330Ω
12kΩ
10μF
63V
D23
1N4004
47μF
63V
100nF
D1
1N5819
2.7kΩ
10μF
9V DC
INPUT
CON1
* UF4002
can be used
instead
47μF
D3 1N5819*
10Ω
−18V
D26 1N4004
+14V
REG
D9
1N5819*
2.2Ω
47μF
100nF
220μF
63V
+18V
REG3 LM317
390Ω
D27
1N4004
47μF
L1 100μH
D2 1N5819*
10Ω
+18V
100nF
3.9kΩ
10μF
0V
220μF
25V
100nF
IC4
33kΩ
220μF
63V
100nF
10μF
2.4kΩ
2.7kΩ
1μF
3.9kΩ
10μF
100nF
47μF
390Ω
D28
1N4004
−18V
REG4 LM337
D29 1N4004
−14V
REG
LM2577T
1
2
3
4
5
LM317T
ADJ
OUT
IN
LM337T
ADJ
IN
OUT
1N5819, UF4002,
1N4004
A
K
SC
©2024
MICROPHONE PREAMPLIFIER
POWER SUPPLY SECTION

Fig.5: the power supply circuitry uses two switch-mode regulator ICs, one charge pump and three adjustable linear regulators to generate a 48V DC phantom power rail plus regulated ±14V rails for the op amps. Those are all derived from a single 9V DC input.

final regulated rails will still be close to ±14V. While the negative rail can only provide a modest current, we only need about 40mA total to power a few op amps.

PCB layout

We have laid the board out so that it is a neat, if tight, fit into a standard 120 × 93 × 35mm diecast aluminium enclosure. It is just large enough to accommodate the PCB, two XLR connectors and the switches, but small enough not to get in your way in use. The aluminium is tough enough to take some abuse without getting ratty or cracking.

Due to the fairly packed board, it was important to put the switch-mode regulators at one end and the preamp circuitry at the other and use extensive ground planes to keep the noise down. The resulting board configuration is shown in Fig.6.

To get it to fit, we had to lay the board out with cutouts for the integral pillars in the corners of the enclosure and a cutout into which the XLR connectors sit. That allowed us to use through-hole parts exclusively, so it's straightforward for anyone to build.

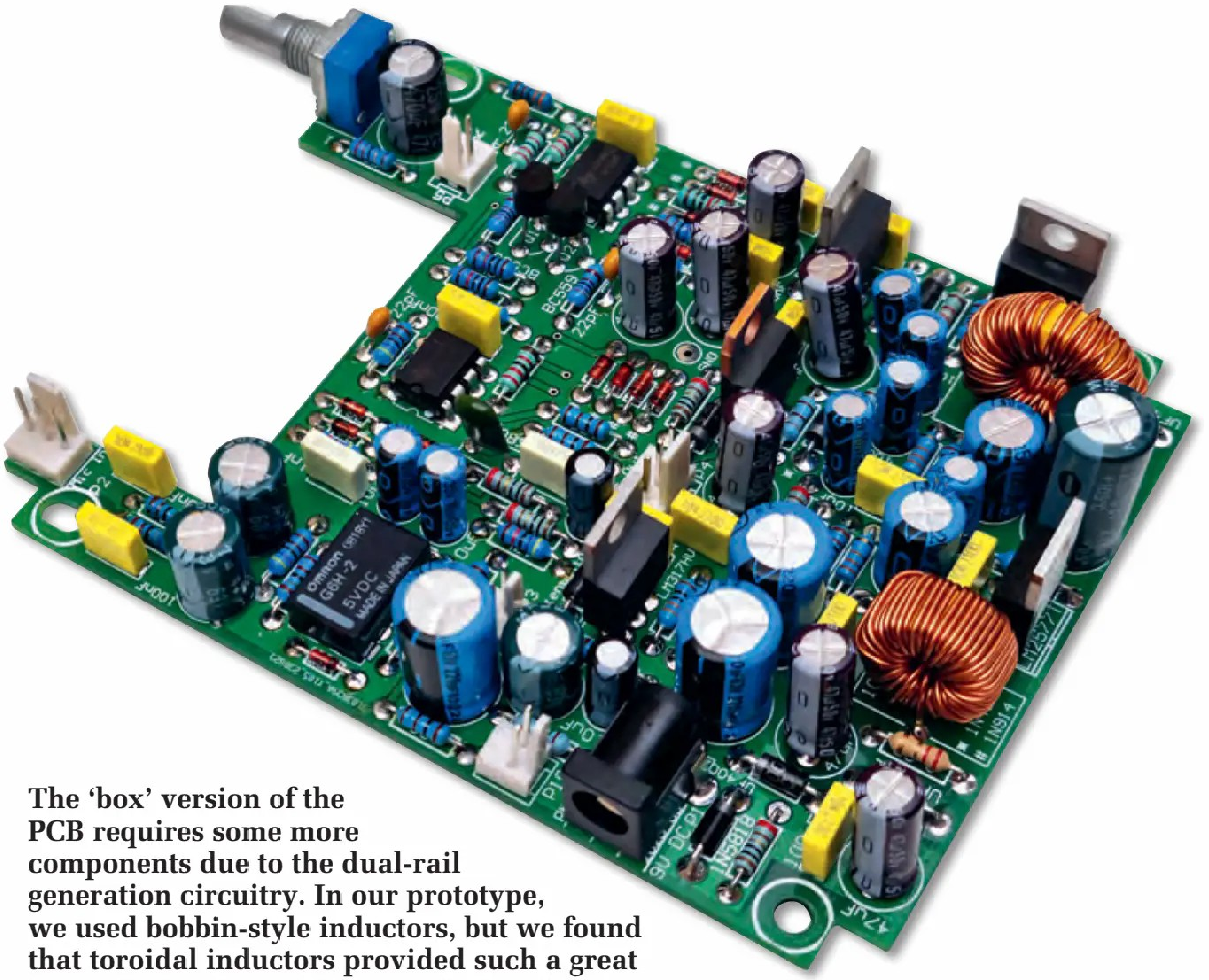
Suppose you are integrating this into a larger enclosure, such as an existing preamp. In that case, we have designed a separate 'embedded' version of the board without the LM2577 that generates the positive and negative rails (IC4). That means you can run it from external ±15V rails instead. At the same time, we filled in the cutout as it would serve no purpose in such an application.

Everything else is basically identical, so you can use the same overlay diagrams regardless of which version you build. Just leave out the parts that don't exist on the embedded version (in case that is not obvious!). When buying your board, make sure you choose the version that suits your needs.

The only other difference in components is that the 150Ω resistor next to CON10 is increased to 330Ω and two of the 3.9kΩ resistors have been reduced to 3.0kΩ so that the LM317/337 regulators will not go into dropout with their inputs at ±15V rather than the ±18V generated by the switching regulator in the other design.

Construction

The Microphone Preamp is built on a double-sided PCB coded either



The 'box' version of the PCB requires some more components due to the dual-rail generation circuitry. In our prototype, we used bobbin-style inductors, but we found that toroidal inductors provided such a great improvement in performance that we had to change them to the design presented.

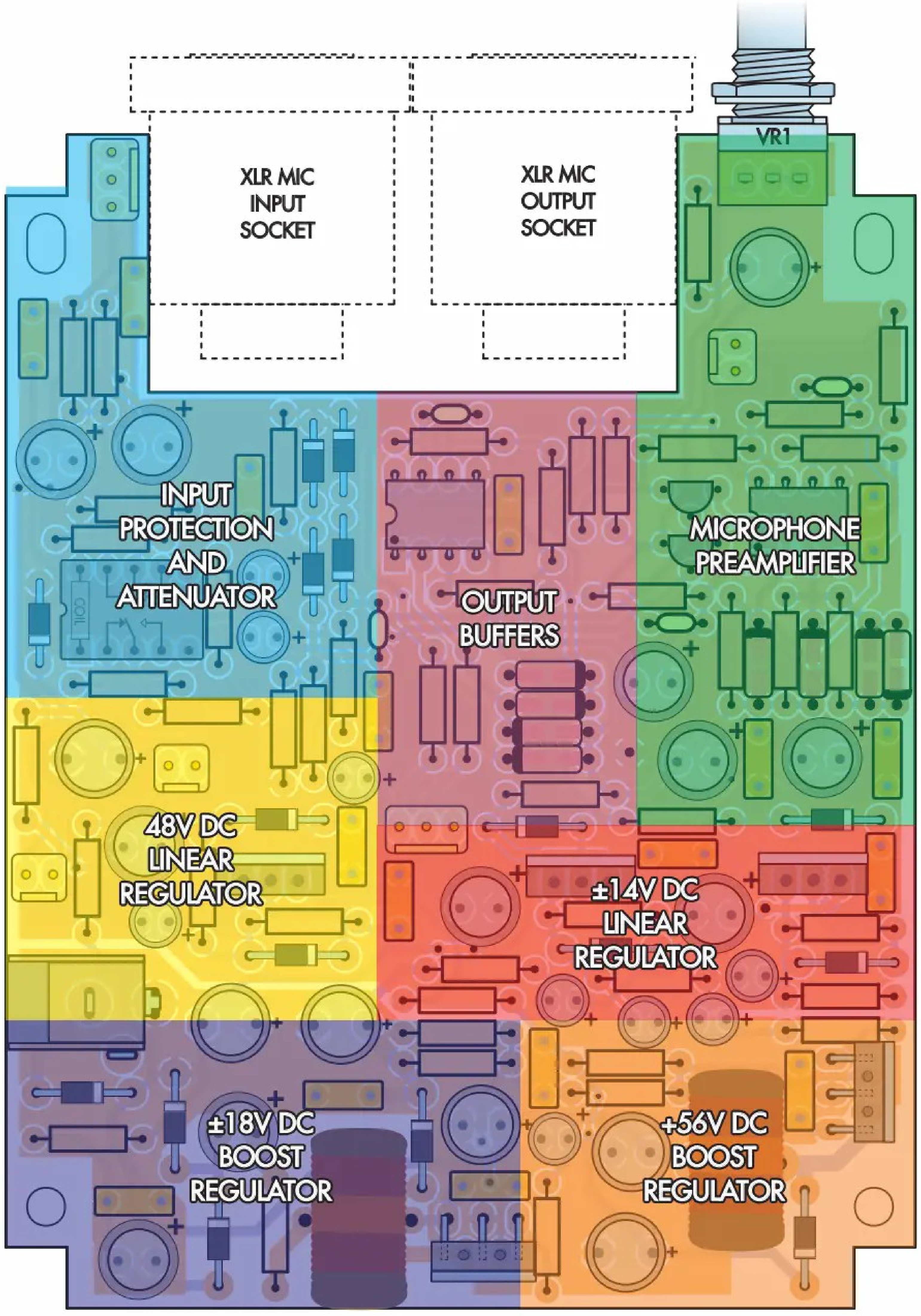


Fig.6: how the various circuit sections have been arranged on the PCB. This configuration allows it to fit in a compact case while keeping the noisy switch-mode ICs away from the sensitive analog preamplifier circuitry.

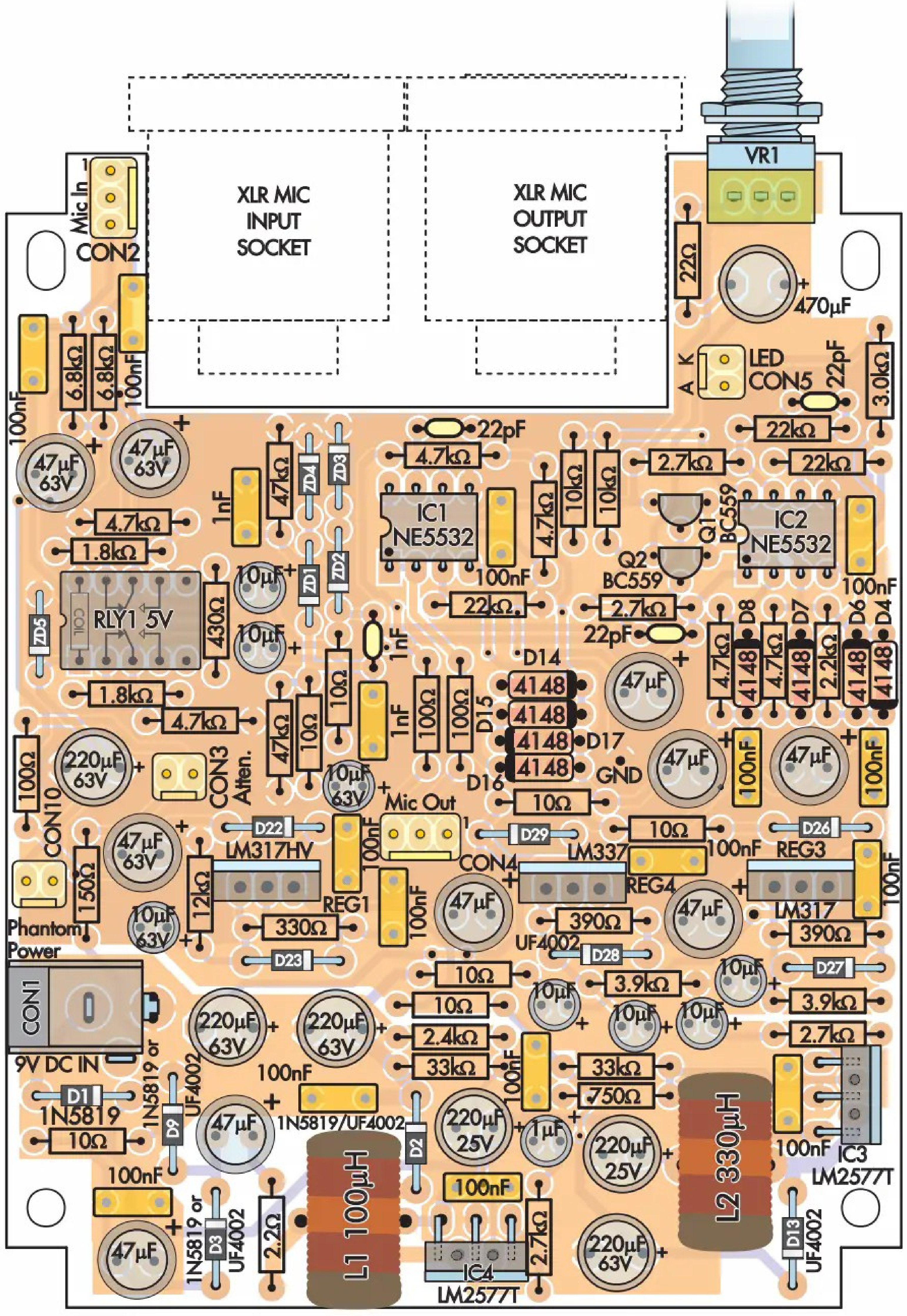

XLR MIC INPUT SOCKET
XLR MIC OUTPUT SOCKET
VR1
Mic In
CON2
470µF
22Ω
LED
CON5
22pF
3.0kΩ
A K
6.8kΩ
6.8kΩ
100nF
47µF 63V
4.7kΩ
1.8kΩ
RLY1 5V
IC1 NE5532
IC2 NE5532
Q1 BC559
Q2 BC559
22kΩ
2.7kΩ
10kΩ
ZD1
ZD2
ZD3
ZD4
ZD5
430Ω
1nF
10µF
D14
D15
D16
D17
4148
GND
CON3 Atten.
Mic Out
220µF 63V
100Ω
CON10
150Ω
Phantom Power
LM317HV
REG1
CON4
LM337
REG4
REG3
LM317
D22
D23
D26
D27
D28
D29
330Ω
390Ω
UF4002
3.9kΩ
10Ω
2.4kΩ
33kΩ
750Ω
CON1
9V DC IN
D1 1N5819
1N5819 or UF4002
D2
D3
D9
1N5819/UF4002
L1 100µH
L2 330µH
220µF 25V
IC3 LM2577T
IC4 LM2577T
D13 UF4002
2.2Ω
2.7kΩ
100nF

Fig.7: this version of the PCB suits the diecast metal case, with a cutout at the top for the XLR sockets to fit. The diodes, electrolytic capacitors and ICs are all polarity sensitive, so make sure they are orientated as shown here.

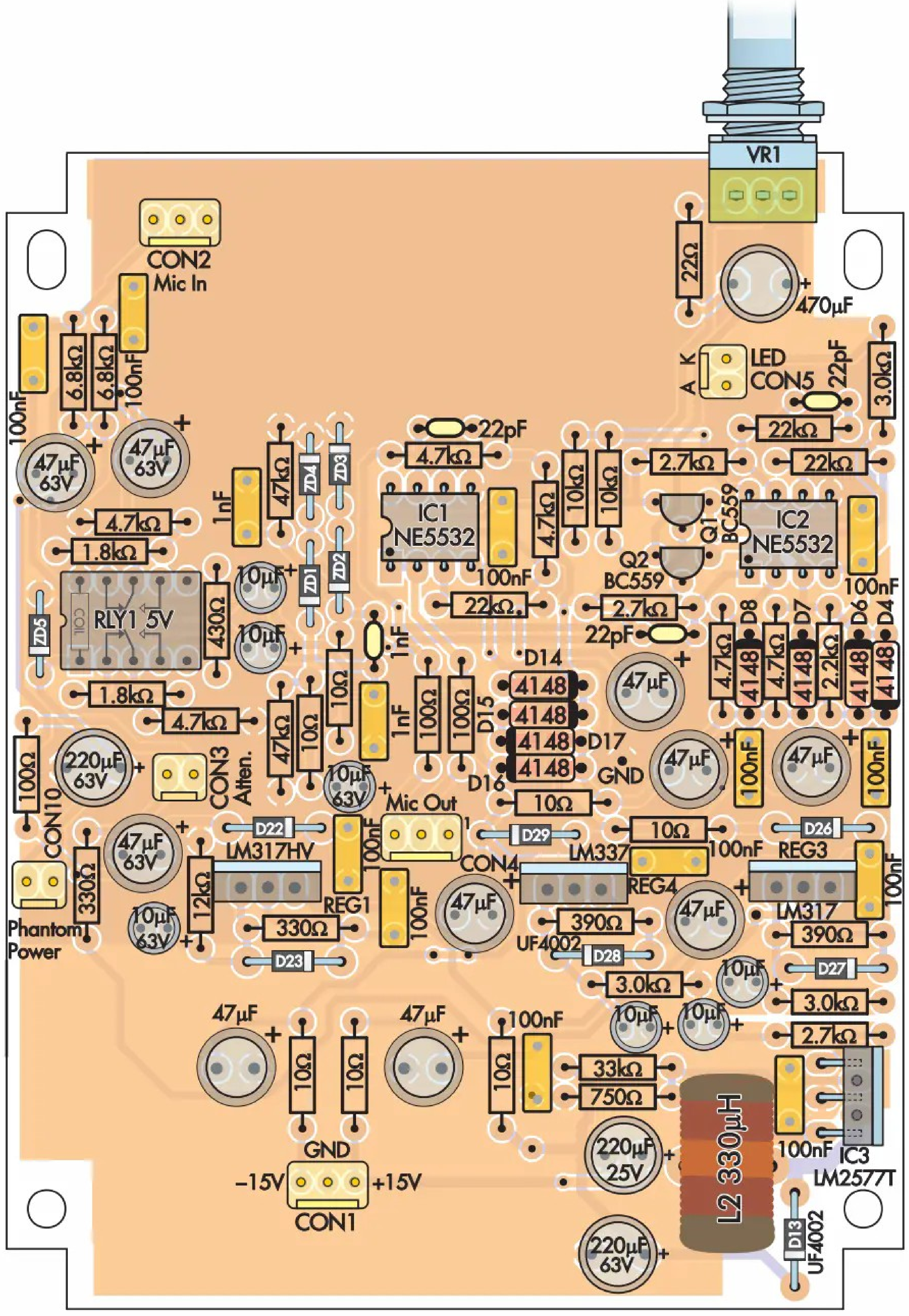

VR1
CON2
Mic In
470µF
22Ω
LED
CON5
22pF
3.0kΩ
6.8kΩ
100nF
47µF 63V
4.7kΩ
1.8kΩ
RLY1 5V
IC1 NE5532
IC2 NE5532
Q1 BC559
Q2 BC559
22kΩ
2.7kΩ
10kΩ
4148
D14
D15
D16
D17
GND
CON3 Atten.
Mic Out
220µF 63V
CON10
330Ω
Phantom Power
LM317HV
REG1
CON4
LM337
REG4
REG3
LM317
UF4002
390Ω
3.0kΩ
47µF
10Ω
33kΩ
750Ω
GND
−15V
+15V
CON1
L2 330µH
220µF 25V
IC3 LM2577T
D13 UF4002

Fig.8: the 'embedded' version of the PCB removes the split rail generators so it can run from ±15V DC rails (or similar) that might already be available within a mixer, preamplifier or power amplifier.

01110231 (full version) or 01110232 (embedded version) and measuring 85 × 110mm in either case. The two layouts are shown in Figs.7 & 8. The main difference is the omission of the dual rail generation circuitry in the 'embedded' version. Most other parts and locations remain the same.

We will describe building the full PCB that fits in the small case. You simply skip the missing parts for the embedded version that operates from dual rails. The only added part is the three-pin header for power input CON1 rather than the barrel socket.

Start by fitting all the resistors. The pairs of 6.8kΩ, 4.7kΩ and 1.8kΩ resistors in the input section at upper left, need some care. These parts should ideally be matched to better than 1%; we bought 10 of each and chose the two that measured the closest for each pair. That improves the common mode (noise) rejection.

Now move on to the diodes. There are five different diode types, so don't get them mixed up and ensure that the cathode stripes are orientated as shown in the overlay diagrams. Note that the 400mW zener diodes look similar to the 1N4148 small signal diodes, so be careful with those. While diodes D2, D3 & D9 can be either UF4002 or 1N5819 high-speed types, D13 must be a UF4002.

Next, mount all the non-polarised capacitors, ie, the ceramic and plastic film types. Follow with the electrolytic capacitors, which are polarised. They all face the same way, with the positive (longer) lead to the right and the stripe on the can to the left.

We have marked the 63V-rated capacitors on the PCB, although if you use the parts specified in the parts list, they will already have the correct ratings.

Now install the power socket, two-pin and three-pin polarised headers, the two toroidal inductors (which are not polarised) and the potentiometer. The orientations of the polarised headers are not critical, but if you use our suggested orientations, you're less likely to make mistakes following our wiring instructions.

We can now fit the two LM2577s and test the boost regulators. Depending on whether yours come with staggered or straight leads, you might need to bend the leads to fit the pads. Ensure that the regulators sit close to the PCB and do not hang off the edge. You can put a dab of neutral cure silicone under the inductors.

Initial testing

To test the switching part of the board, connect a 9V DC plugpack and check the voltages on either side of D1, the protection diode. There should be 9V on the anode and over 8.5V on the cathode. If not, check for shorts and things getting hot, and verify that your plugpack has negative on the tip and positive on the ring (the opposite of many that you'll find).

Check the voltage on either end of the 10Ω resistor immediately next to the 33kΩ resistor (it's all by itself on the embedded version, to the left of that 33kΩ resistor). You should get readings at both ends of 55V ±5V. Do not touch this with your fingers as it is a high enough voltage to bite. If that is not correct, check the parts in the lower-right corner, especially IC3, and verify the orientation of D13.

For the dual rail voltage generator on the non-embedded version, measure the voltage on either end of two more 10Ω resistors in the power supply section. One is just to the left of D26, while the other is just above D29. These should be ±18.4V ±1.5V. Again, if these voltages are not correct, stop and work out why. The likely culprit is incorrect diode or capacitor orientation.

If IC3 or IC4 is not working, put a scope probe on pin 4 of IC4. You should see a switching waveform at around 52kHz. If not, it might not be getting power.

Now fit the LM317HV, LM317 and LM337 devices (REG1, REG3 & REG4). After that, check the voltage on CON10, the phantom power header. It should be 48V ±4V. Also check the voltage on pins 4 and 8 of the (still empty) IC1 and IC2 locations. You should measure +14V ±1V on pins 8 and -14V ±1V on pins 4.

Again, if one of these is off, there must be a problem around the associated regulator, so check the input voltages, and the orientations of the regulators and associated protection diodes.

With the power supply now fully operational, mount the relay (watch its orientation), the two BC559 transistors and the two NE5532 op amps, which can be soldered directly to the board or socketed (although using sockets could reduce its robustness). Double-check their orientation before soldering, as desoldering op amps or relays is hard. If you have to remove one, cut off all the legs and desolder them individually.

Re-apply power and check that the relay works by shorting the pins of CON3; you should hear the relay click. If not, check that the relay is the right way around and that you have ZD5 orientated correctly.

You can now plug in a microphone or oscillator, with a maximum input level of 100mV, to the CON2 input and check that it is amplifying the signal correctly and delivering correct output signals at the pins of CON4.

If you don't get an output, check that you have phantom power on if required. Place a shorting block across CON10 if necessary. There should be close to 48V on the CON10 pins and a

Parts List – Compact Microphone Preamplifier

1^{0} double-sided PCB coded 01110231, 85 × 110mm
1 120 × 93.5 × 35mm diecast aluminium box
[Hammond 29830PSLA; Mouser 546-29830PSLA, Bitsbox EN013 etc]
1^{0} 9V DC 700mA+ plugpack with 2.1mm ID plug
1^{0} 100μH toroidal inductor (L1) **[eg, Würth 7447021, Farnell 2082528]**
1 300-330μH toroidal inductor (L2) **[eg, Würth 7447060, Farnell 2082535]**
1 9mm 10kΩ reverse log potentiometer (VR1)
[Mouser 858-P091NFC25CR10K or 652-PTD9012015FC103]
1 knob to suit VR1 (D shaft), around 13mm in diameter
1^{0} PCB-mounting 2.1mm inner diameter barrel socket (CON1)
2 8-pin DIL IC sockets (**optional**; for IC1 & IC2)
2 3-pin polarised headers, 2.54mm pitch, with matching plugs and pins (CON2, CON4)
3 2-pin polarised headers, 2.54mm pitch, with matching plugs & pins (CON3, CON5, CON10)
1 3-pin female chassis-mount XLR socket (CON11)
1 3-pin male chassis-mount XLR socket (CON12)
2 SPDT chassis-mount mini toggle switches (S1, S2)
1 5V DC coil DPDT PCB-mounting telecom relay (RLY1) **[eg, EA2-5NU, Farnell 2766136]**
1 panel-mount green 3mm LED with bezel (LED1)
8 M3 × 16mm panhead machine screws
4 6mm-long M3-tapped Nylon spacers
10 M3 shakeproof washers
6 M3 hex nuts
4 stick-on rubber feet
3 1m lengths of light-duty hookup wire (eg, white, red & black)
1 short length of 3mm diameter heatshrink tubing

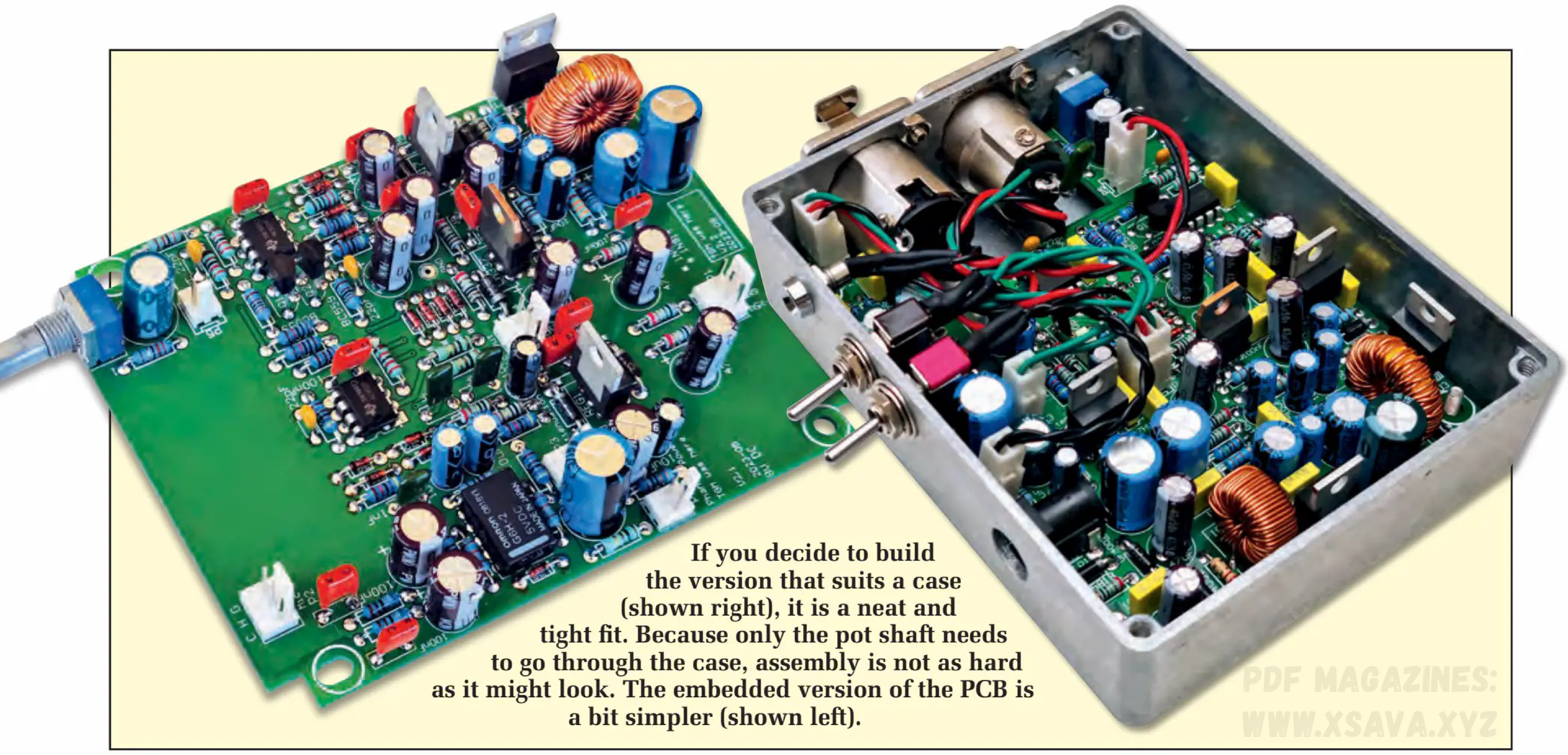
Semiconductors
2 NE5532 dual low-noise op amps, DIP-8 (IC1, IC2)
2^{1} LM2577T integrated switch-mode regulators, TO-220-5 (IC3, IC4)
1 LM317HV (preferred) or LM317 adjustable linear regulator, TO-220-3 (REG1)
1 LM317 adjustable linear regulator, TO-220-3 (REG3)
1 LM337 adjustable negative linear regulator, TO-220-3 (REG4)
2 BC559 low-noise PNP transistors, TO-92 (Q1, Q2)
5 6.8V 400mA axial zener diodes, DO-35 (ZD1-ZD5)
1^{0} 1N5819 40V 1A schottky diode, DO-41 (D1)
3^{0} 1N5819 40V 1A schottky or UF4002 100V 1A ultrafast diodes, DO-41 (D2, D3, D9)
1 UF4002 100V 1A ultrafast diode, DO-41 (D13)
8 1N4148 75V 200mA diodes, DO-35 (D4, D6-D8, D14-D17)
6 1N4004 400V 1A diodes, DO-41 (D22, D23, D26-D29)

Capacitors
1 470μF 25V radial electrolytic; 5mm pitch, maximum 10mm diameter & 21mm high
4^{2} 220μF 63V radial electro; 5mm pitch, maximum 10mm diameter & 21mm high ◆
2^{1} 220μF 25V radial electro; 3.5mm pitch, maximum 8mm diameter ◆
10 47μF 63V radial electro; 2.5-3.5mm pitch, maximum 8mm diameter & 21mm high
8^{7} 10μF 63V low-ESR radial electrolytic
1^{0} 1μF 50V/63V radial electrolytic
15^{12} 100nF 63V/100V MKT
1 10nF 63V/100V MKT
3 1nF 63V/100V MKT
3 22pF 50V C0G/NP0 ceramic
◆ *low-ESR types are preferred but not required*

Resistors
2 47kΩ
2^{1} 33kΩ
3 22kΩ
1 12kΩ
2 10kΩ
2 6.8kΩ
6 4.7kΩ
2^{0} 3.9kΩ
1^{3} 3.0kΩ
1^{0} 2.4kΩ
4^{3} 2.7kΩ
1 2.2kΩ
2 1.8kΩ
1 750Ω
1 430Ω
2 390Ω
1^{2} 330Ω
1^{0} 150Ω
3 100Ω
1 22Ω
7 10Ω
1^{0} 2.2Ω

For the embedded version, add:
1 double-sided PCB coded 01110232, 85 × 110mm
1 3-pin polarised header, 2.54mm pitch, with matching plugs and pins (CON1)
^{num} *digit indicates how many to use for the embedded version*

SILICON CHIP Microphone Preamp Kit (SC6784, $84 [~£43] incl. VAT + P&P): includes the standard PCB plus all onboard parts, switches and mounting hardware. The case, XLR connectors, bezel LED and wiring are not included.


If you decide to build the version that suits a case (shown right), it is a neat and tight fit. Because only the pot shaft needs to go through the case, assembly is not as hard as it might look. The embedded version of the PCB is a bit simpler (shown left).

PDF MAGAZINES: WWW.XSAVA.XYZ

'reasonable' DC voltage at pins 2 and 3 of the input connector. This will vary depending on the microphone; expect it to be between about 5V and 43V.

If you still have trouble, use an oscillator to drive the 'hot' input (middle pin of CON2) and:

- Check the input voltage with a scope. It should be set to 100mV.
- Check the voltages on the 1.8kΩ resistors immediately on either side of RLY1. Assuming the use of a single-ended oscillator, one of these should have your test voltage. Switch the attenuator in and out; you should see a 20dB (10 times) reduction in voltage level at one end.
- Check the base voltages of Q1 & Q2. They should be about 0V (a 'touch' above, to be precise!)
- Check the Q1 & Q2 emitter voltages; they should be about 0.6V.
- There should be about 10V across the two 10kΩ resistors right next to the XLR socket cutout, on either side of the 470µF capacitor, and about 4.7V across the 4.7kΩ resistors immediately to the right of D7 and below D8. Check the orientations of D7 and D8 if these voltages are not right. These voltages should be identical as they connect to the inverting and non-inverting inputs of the same op amp (IC2a).
- Check the voltage on pin 1 of IC2; it should be close to 0V with no signal applied to the Preamp. If it is pegged to one of the supply rails, look for something amiss in the feedback loop through IC2a, IC2, Q1 & Q2.

If it's working, check that the gain control provides about 48dB of range. You will need to drop the input voltage at high gain settings to avoid clipping. You should be able to achieve more than 8V RMS between pins 2 and 3 of the output connector into a 600Ω load.

Case preparation

The 120 × 93.5 × 35mm (119mm from some sources) diecast enclosure is available from a range of suppliers. All our measurements assume the use of 6mm standoffs for mounting the PCB, which provide clearance for the attenuation and phantom power switches and taller low-ESR capacitors. If you want to use different standoffs, verify that everything will fit, especially the 63V capacitors and switches. Standoffs taller than about 8mm are unlikely to work.

Start by drilling and deburring the holes in the side walls of the enclosure, as shown in Fig.9; hold off on the mounting holes in the base.

We used a stepped drill bit to make the XLR connector holes. These are a real boon for making larger holes. We bought several types of XLR connectors and found they were all similar

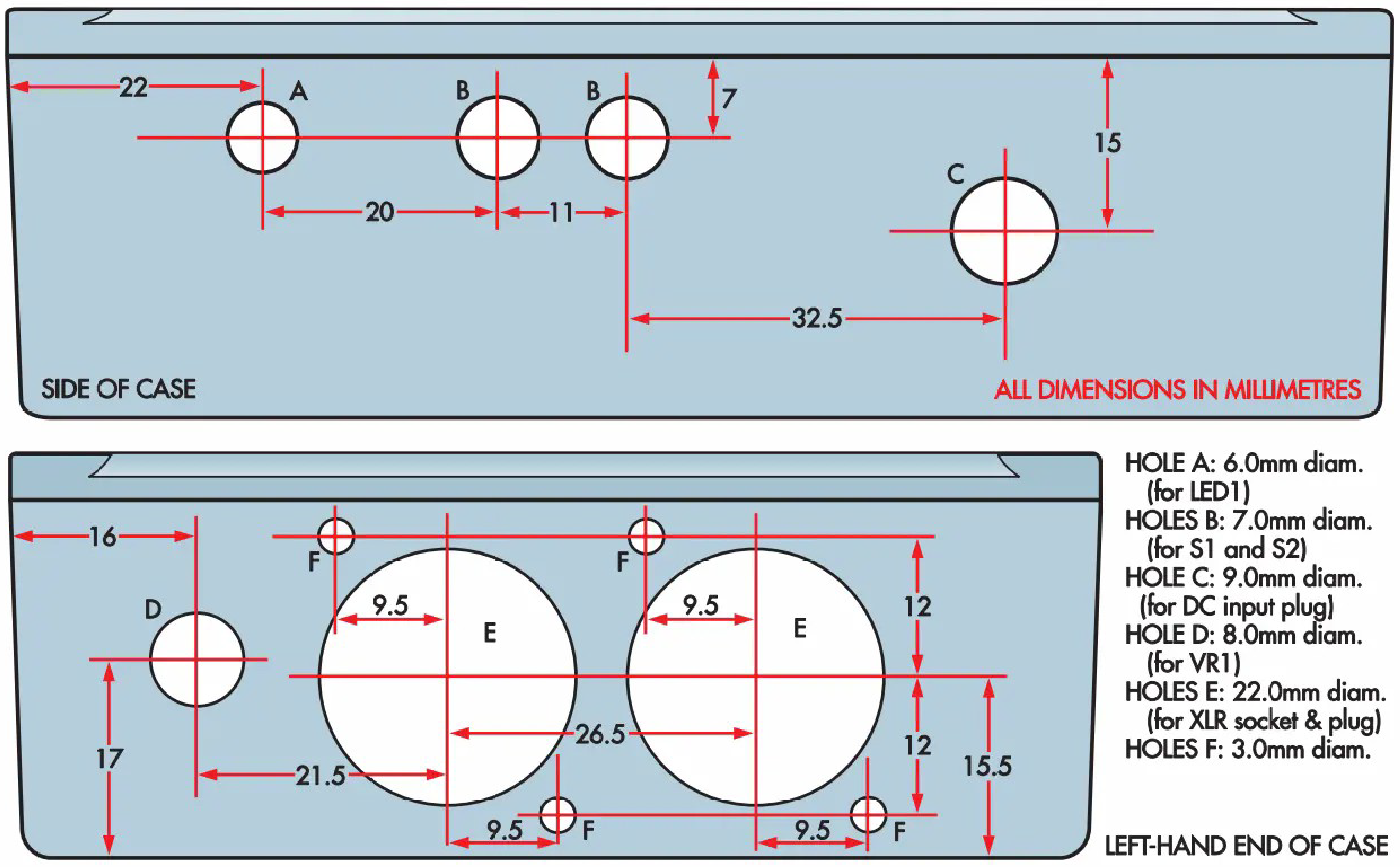

22
A
B
B
7
15
C
20
11
32.5
SIDE OF CASE
ALL DIMENSIONS IN MILLIMETRES
16
F
F
D
9.5
E
9.5
E
12
12
15.5
17
26.5
21.5
9.5
F
9.5
F
HOLE A: 6.0mm diam. (for LED1)
HOLES B: 7.0mm diam. (for S1 and S2)
HOLE C: 9.0mm diam. (for DC input plug)
HOLE D: 8.0mm diam. (for VR1)
HOLES E: 22.0mm diam. (for XLR socket & plug)
HOLES F: 3.0mm diam.
LEFT-HAND END OF CASE

Fig.9: the drilling details for the XLR sockets and holes for the potentiometer, LED, DC socket and switches. Leave the small XLR mounting holes until you have the sockets ready to install so you can position them accurately.

but differed in the required cutout. You might need to fine-tune your metalwork for your connector.

We also recommend that you hold off drilling the smaller fixing holes for the XLR connector until after you have made the main hole. Once the connector fits OK, mark and drill these holes so they are in the ideal locations.

The two lower holes for the XLR connectors will need to be drilled and tapped for a 3mm thread (drill to 2.5mm first), as there is no room for nuts inside the case. An alternative is to use a long 3mm pop rivet, an approach we have tried and found to work well, especially if you get a hole slightly crooked.

Once you have the side holes drilled, present the PCB to the case without the standoffs attached, and mark the locations of the mounting holes. They are shown in Fig.10 but you should use the PCB to mark them more accurately. Drill these to 3.5mm and deburr them. This method is easiest since getting those measurements perfect inside the box is not easy.

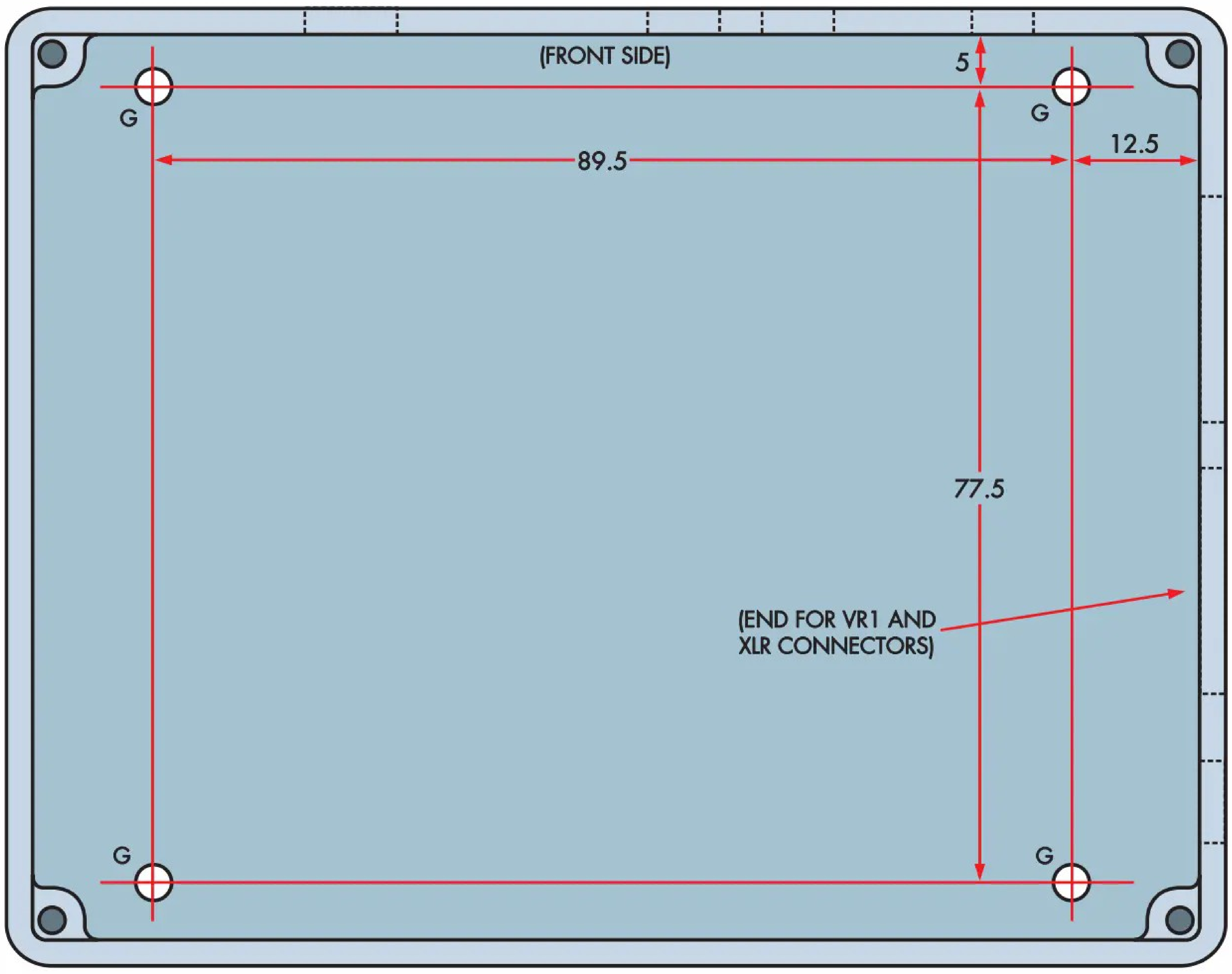
Install the standoffs to the case by putting a 16mm M3 machine screw and M3 shakeproof washer through the panel from the outside, then screw the 6mm standoff onto the machine screw – see Fig.11. Do not fully tighten it, as you need to be able to jiggle the PCB onto the M3 screws. Once the PCB is in place, tighten the screws onto the standoffs. Pushing the PCB onto the standoffs will help you do that.

We placed slotted holes at the connector end of the PCB so you can present the board to the case with the connector end tilted down, allowing the gain control pot shaft to go through the front panel. You can then jiggle the M3 screws through the slotted holes. Once the board is in place, use shakeproof washers and an M3 nut to secure it, as shown in the photos.

Installing the XLR connectors

The input connector is next to the input header, with the output XLR next to the gain control. Solder three differently-coloured 100mm wires to these and twist them together neatly. Trim these back to allow a neat installation, and crimp or solder pins to the pluggable headers. Refer to the wiring diagram, Fig.12, to connect the ground, hot and cold wires to pins 1-3, respectively.

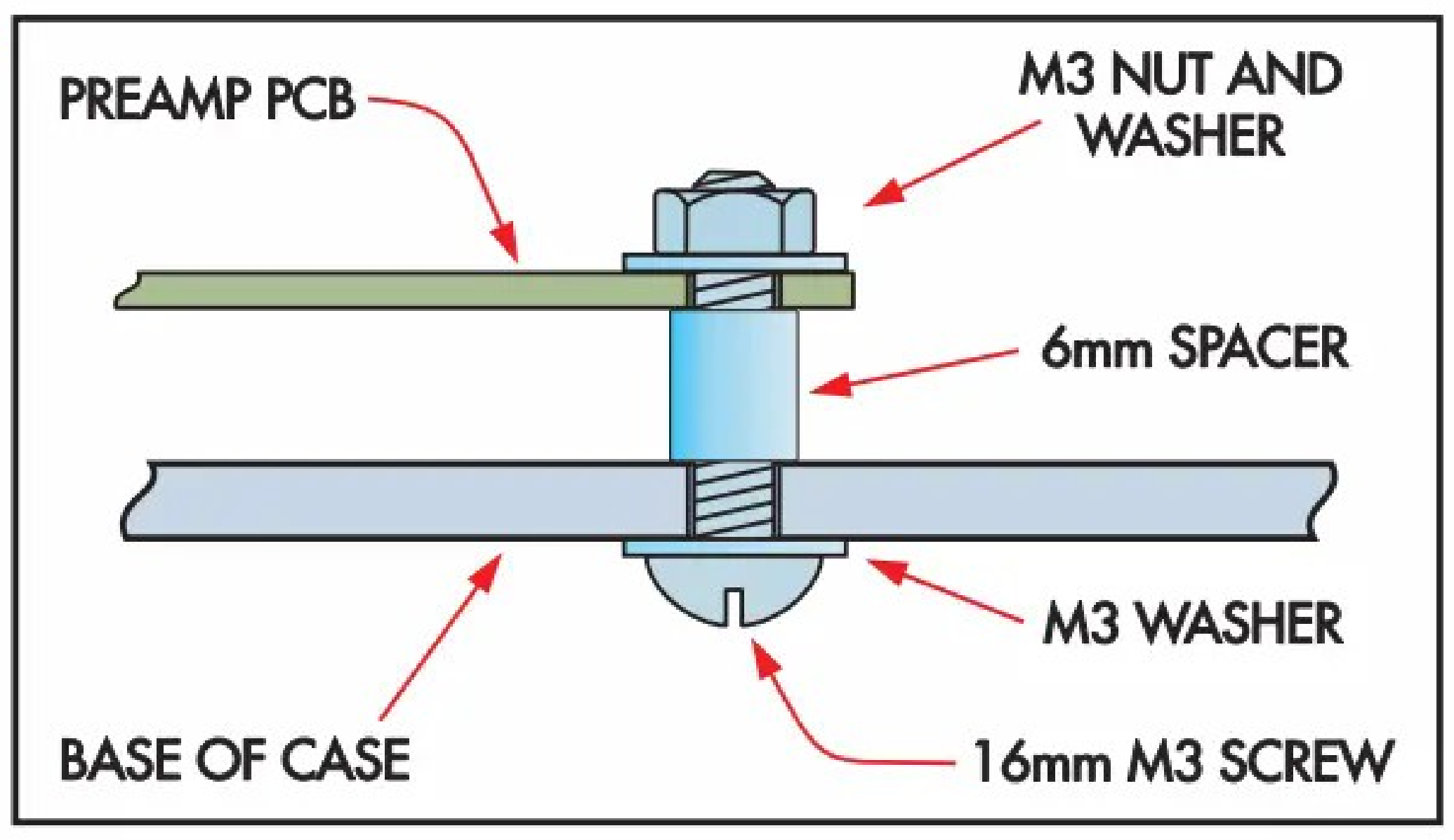
The bottom fixings for the XLR socket



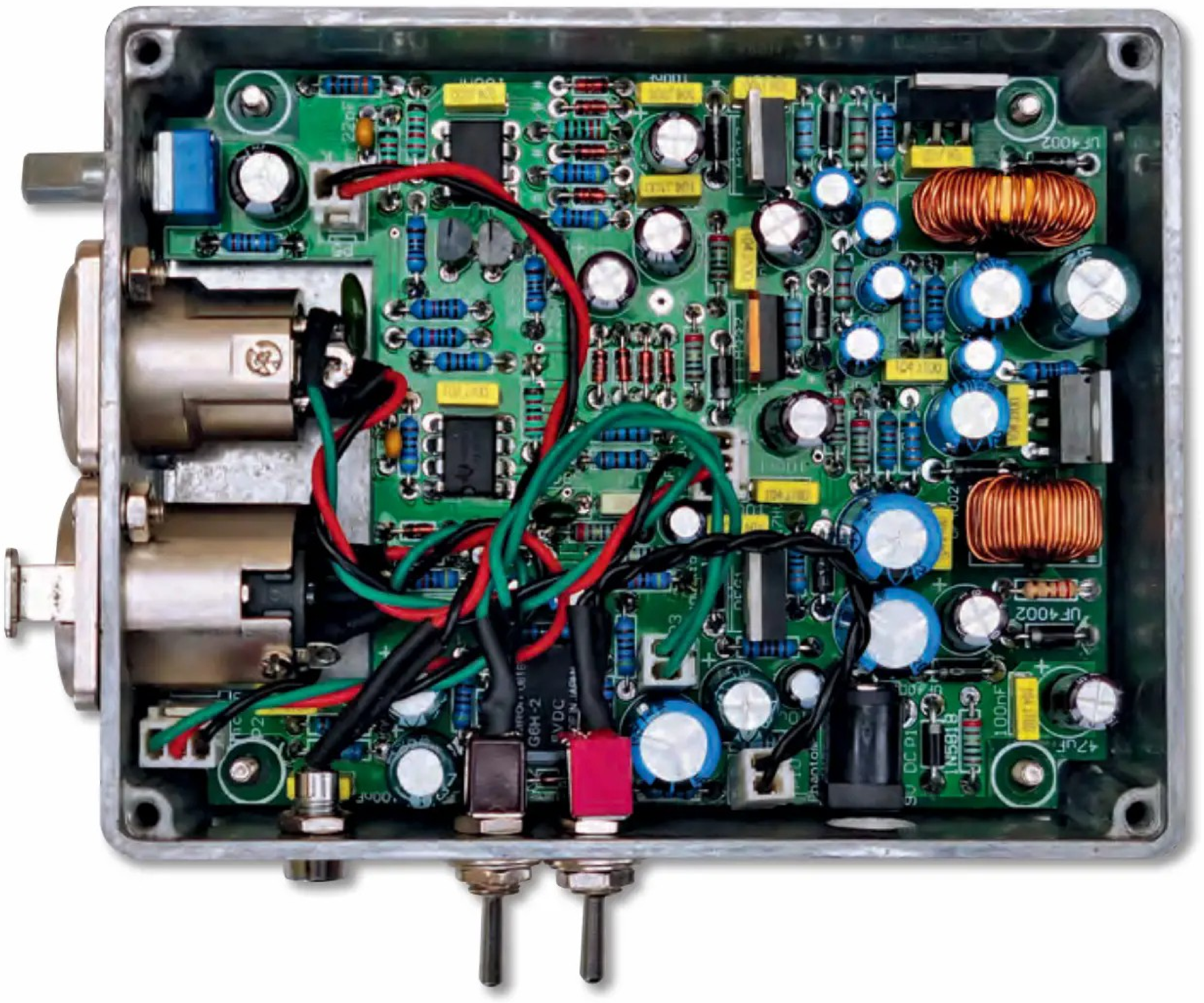
INSIDE OF CASE LOOKING FROM TOP   HOLES G: 3.5mm diameter   ALL DIMENSIONS IN MILLIMETRES

Fig.10: while you can expect the PCB mounting holes to be in these positions, you should use the PCB assembly to mark them exactly before drilling them to ensure everything will fit.

Fig.11: by attaching the standoffs like this, we get a robust result while also allowing us to finagle it into the case. ►



The PCB is designed to accommodate the XLR connectors and just fit inside the case. The board is a tight fit, but the parts are not squished together too much. ▼



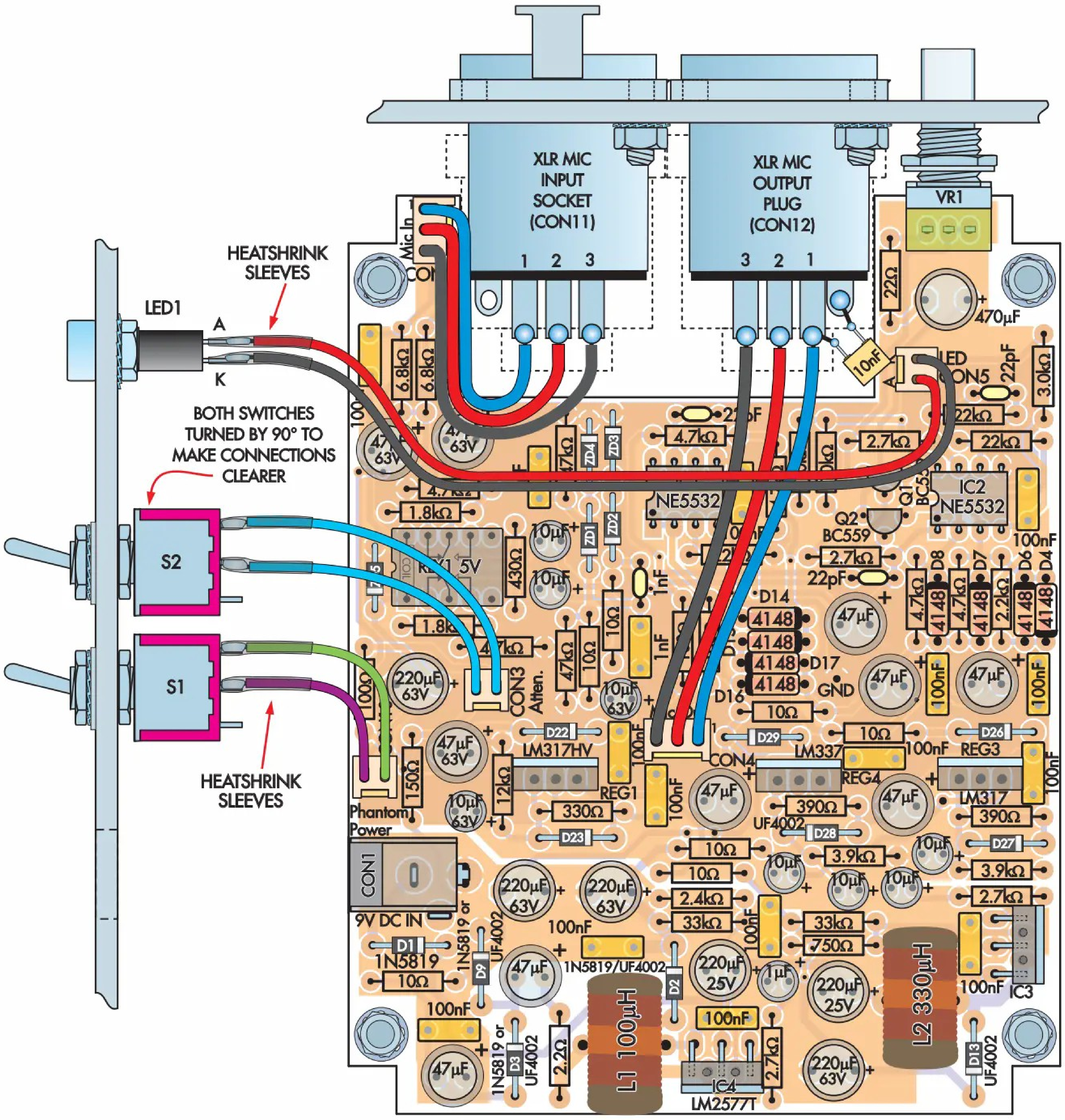

HEATSHRINK SLEEVES
LED1
A
K
BOTH SWITCHES TURNED BY 90° TO MAKE CONNECTIONS CLEARER
S2
S1
HEATSHRINK SLEEVES
XLR MIC INPUT SOCKET (CON11)
1 2 3
XLR MIC OUTPUT PLUG (CON12)
3 2 1
VR1
10nF
LED CON5
Mic In CON
470μF
IC2 NE5532
NE5532
Q2 BC559
Q1 BC5
REG1 LM317HV
REG4 LM337
REG3 LM317
CON3 Atten.
CON4
Phantom Power
CON1
9V DC IN
D1 1N5819
L1 100μH
L2 330μH
IC4 LM2577T
IC3
GND

Fig.12: how to wire it all up. The switches, connectors and LED all connect to the PCB via polarised headers, so you can wire each up one at a time and then plug it all together once the PCB is in the case.

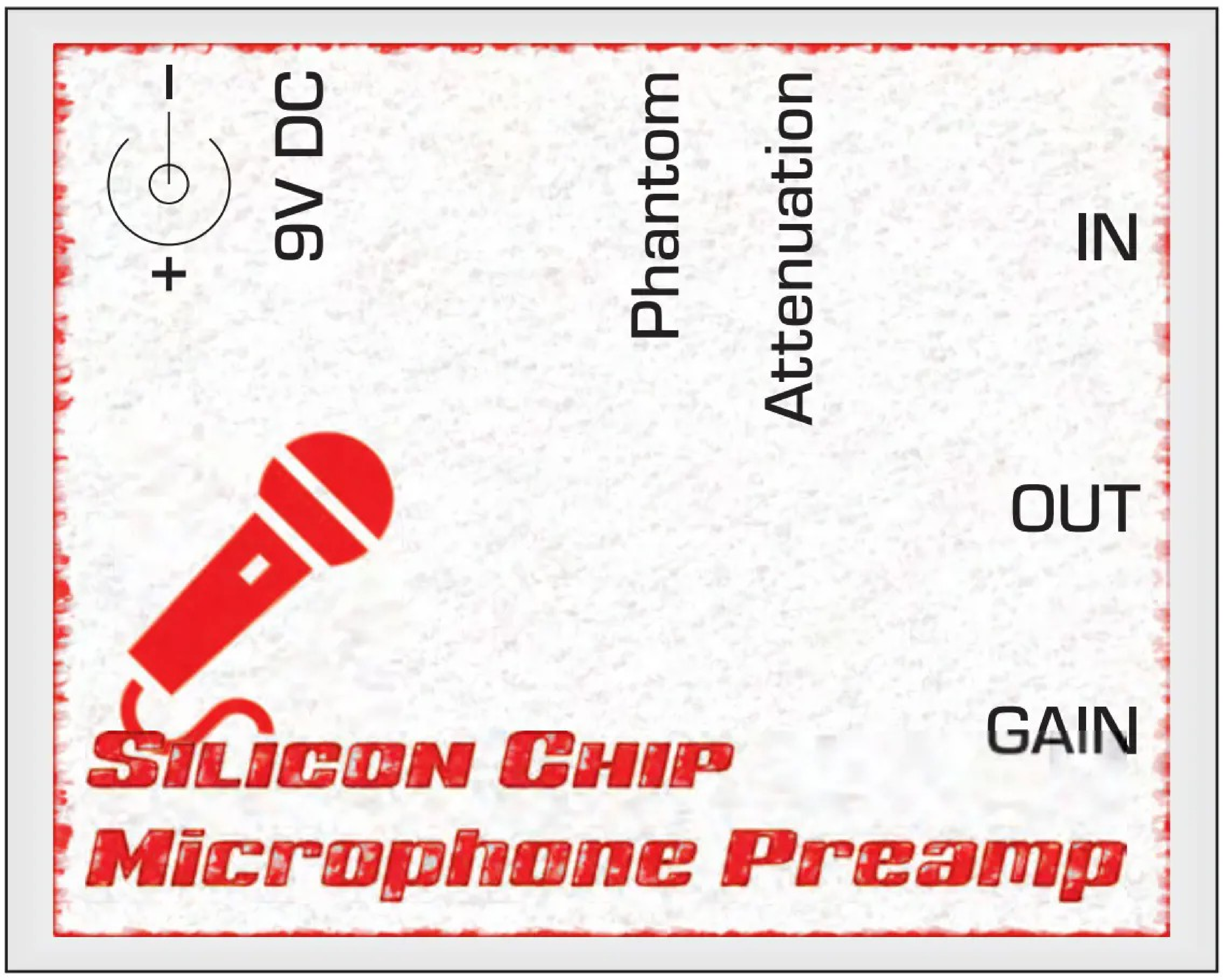

+
–
9V DC
Phantom
Attenuation
IN
OUT
GAIN
SILICON CHIP
Microphone Preamp

Fig.13: print out and attach this lid panel artwork to the top of the box so you (or someone else) will remember what everything does.

are pretty close to the case base, so we simply drilled and tapped ours.

Solder a 10nF capacitor between the case lug on one of the XLR connectors and the ground wire on pin 1. This will effectively ground the case for AC signals.

Switches and LED

The connections for the switches are made with light-duty hookup wire. Use twisted wire (any colour will do) and assemble to the two-pin pluggable headers, as shown in Fig.12.

Similarly, use two pieces of twisted light-duty hookup wire for the LED. Apply heatshrink tubing over the solder connections to it. We used red for the anode and black for the cathode. These connect to pins 1 and 2 of the pluggable header, respectively.

Now attach a knob for the gain control. Make it small, as it will be next to the output XLR connector. You should have tested the board already, so you will be set to go.

We found that the lip of the lid hit the M3 nuts that secure the XLR connectors. To solve that, we used a file to notch the lip on the lid to clear the nuts, and the lid was then a perfect fit.

You will find that the case is very full. The capacitors and TO-220 devices fit with a couple of millimetres of clearance to the lid. We think this is about as good packaging as we could have achieved.

If you are using the ‘embedded’ version, we will leave it to your creativity on where and how you mount the Preamplifier. It is a relatively modest PCB, so it should fit in most places. We would supply the board with ±15V, but you could probably run it from up to ±30V without the regulators getting hot, as the current drain on the linear rails is quite low. You will need to check this detail in your application.

We kept our labelling simple in line with the utilitarian intended use of this device (see Fig.13); you can be creative with this if you wish. Finally, stick some rubber feet on the bottom so it won’t damage the surfaces it’s on and won’t slide around too much.

Using it

The Preamp should generally be run from a 9V DC plugpack. It will work fine from 12V DC. While it will not be damaged by a higher voltage, up to 24V DC, it likely won’t operate as the negative rail will not be generated. PE

Exclusive offer

Win a Microchip Explorer 16/32 Development Kit

Practical Electronics is offering its readers the chance to win a Microchip Explorer 16/32 Development Kit (DM240001-3). Even if you don't win, receive a 15%-off voucher, plus free shipping for one of these products.

The Explorer 16/32 Development Kit is a flexible, convenient and ready-to-start development, demonstration and testing platform for 16-bit PIC24/dsPIC® DSCs and 32-bit PIC32 MCUs from Microchip Technology.

It features all the necessary hardware to begin developing and debugging a complete embedded application. The board accepts Processor Plug-In Modules (PIM) designed for the Explorer 16 or Explorer 16/32 development board for easy device swapping. In addition to the hardware features provided by the board, it can be expanded using PICtail™ Plus daughter cards and mikroBUS™ accessory boards.

It's coupled with the integrated PICkit™-On-Board (PKOB), MPLAB ICD In-Circuit Debugger real-time debug facilities enable faster evaluation and prototyping of application.

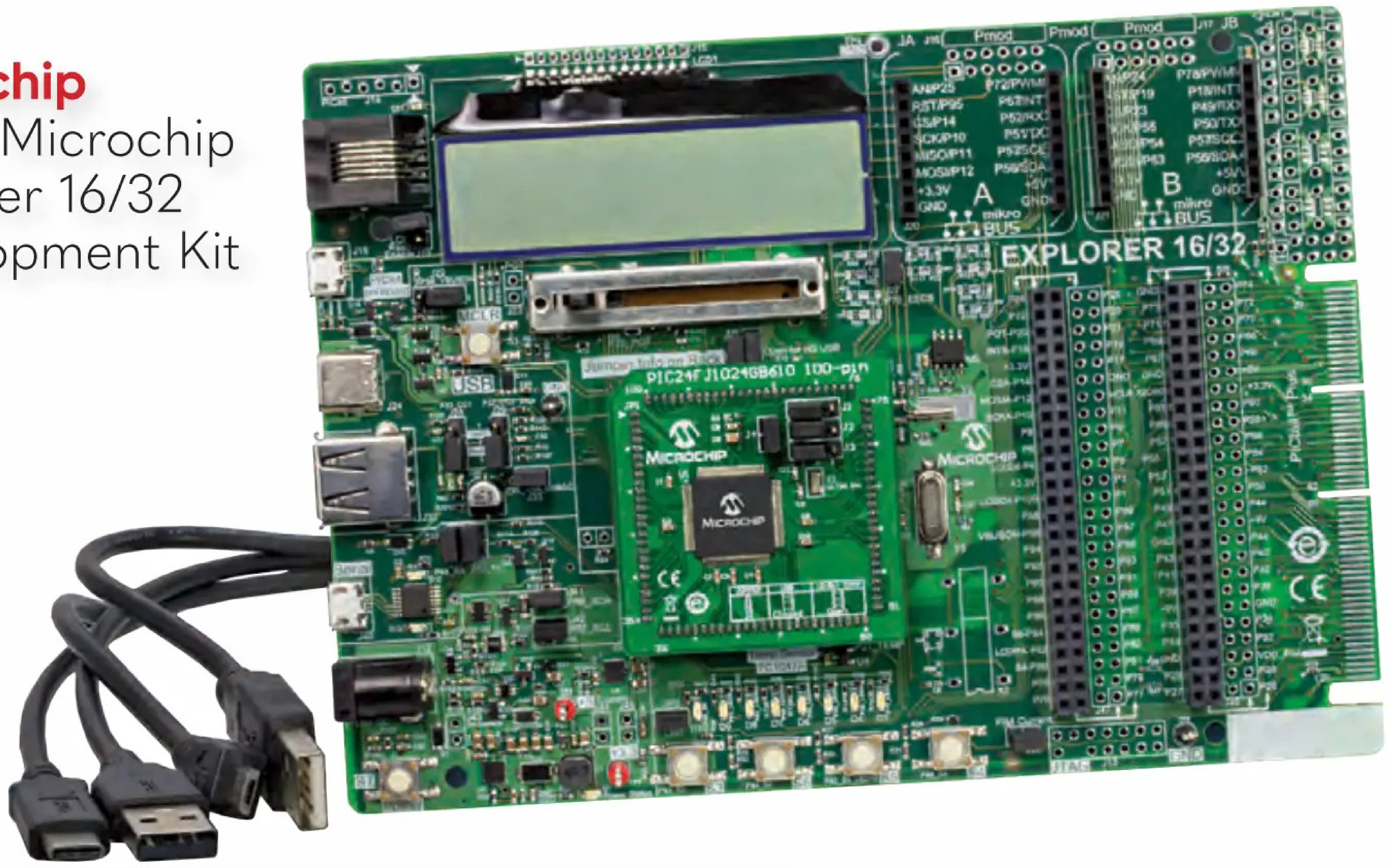
All the Explorer PIMs are supported by this development board. However, not all PIMs are supported by the PKOB. To check the list of supported and unsupported PIMs, refer to the PICkit™ On-Board 3 (PKOB3) Support List. For PIMs not on the PKOB3 support list, use the JP1 or J14 connectors to program the device with a newer generation programming tool.

The development kit comes with the Explorer 16/32 main development board, PIC24FJ1024GB610 PIM (MA240023), USB A to micro-B cable and USB A to Type-C Cable. For out of the box experience, the Explorer 16/32 Development Kit comes with all the necessary components to get started with the evaluation and prototyping right away!

Worth US$161 (about £124) each

Free-to-enter competition

Microchip
Win a Microchip Explorer 16/32 Development Kit



Microchip
DM240001-3

How to enter

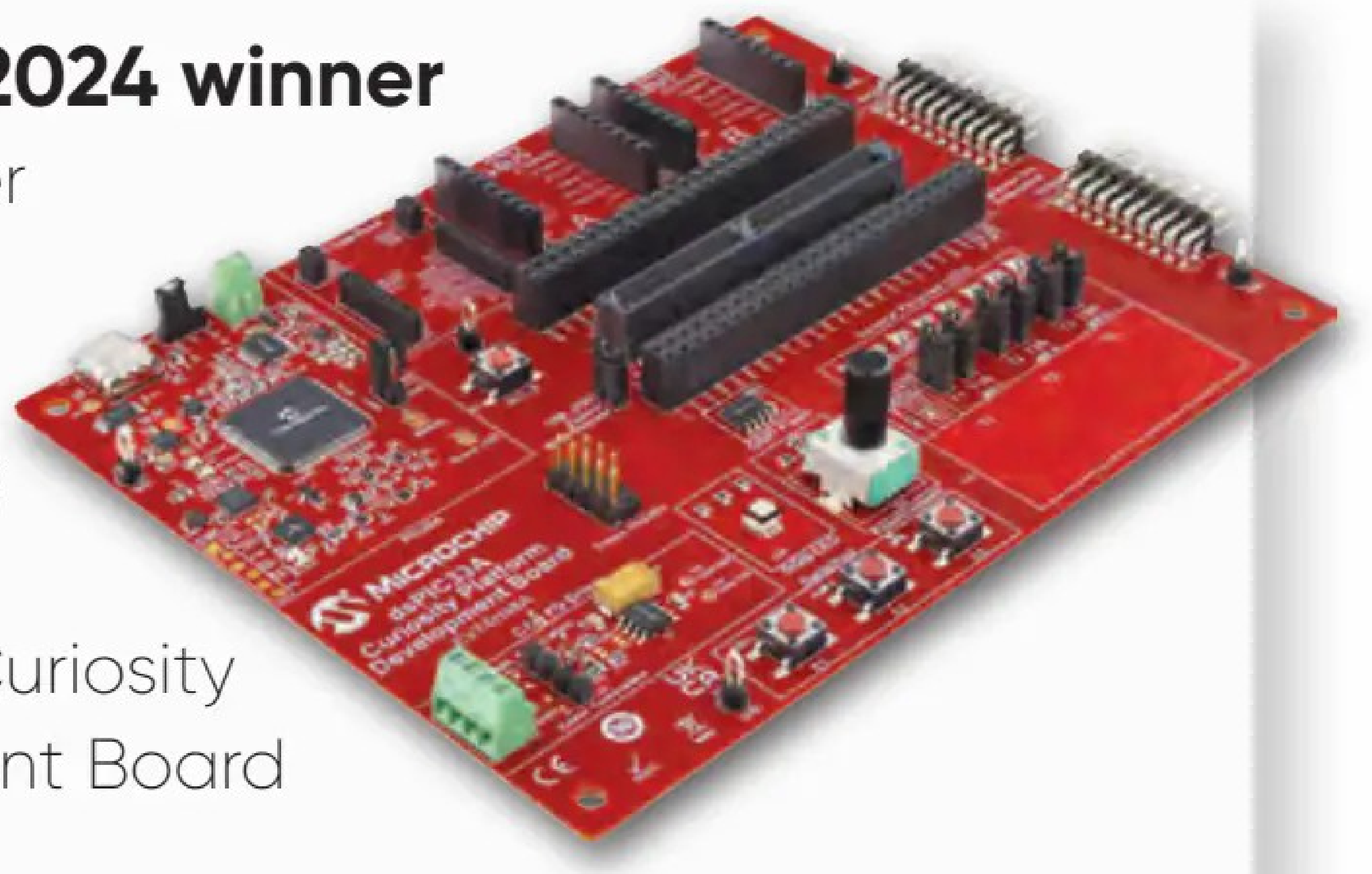
For your chance to win an Explorer 16/32 Development Kit or receive a 15%-off voucher, including free shipping, enter your details in the online entry form at:
https://page.microchip.com/PE-Exp-1632.html

Closing date

The closing date for this offer is 31 December 2024

October 2024 winner

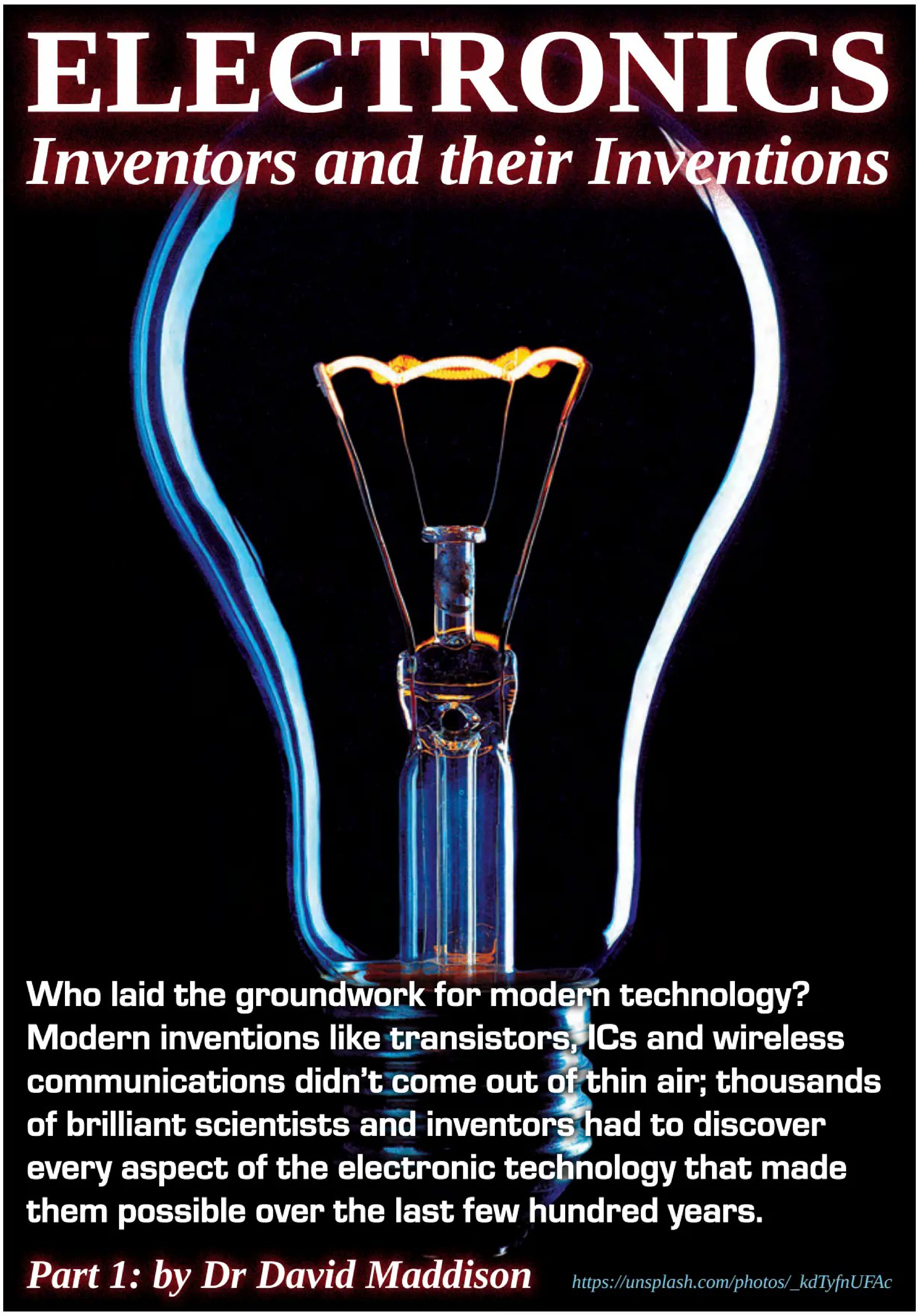
Jack Booker



Jack won a Microchip dsPIC33A Curiosity Development Board

ELECTRONICS

Inventors and their Inventions



Who laid the groundwork for modern technology? Modern inventions like transistors, ICs and wireless communications didn't come out of thin air; thousands of brilliant scientists and inventors had to discover every aspect of the electronic technology that made them possible over the last few hundred years.

Part 1: by Dr David Maddison

https://unsplash.com/photos/_kdTyfnUFAc

We tend to think of electronics as a relatively recent development. Transistors, ICs and many other keystone technologies of the current age were created from the late 1960s to the early 2000s. However, many important discoveries had to be made before any of that was possible.

Modern devices like computer CPUs rely on principles that were discovered hundreds of years ago. We will look at the people behind those discoveries in this series of articles. The inventors and inventions described herein form the basis of all modern electronics.

You may be surprised at how early some advanced concepts were conceived. Many modern devices were invented way ahead of their time. They often failed to find a use then, only to become very popular later.

Many of the scientists and engineers described below contributed far more than we can describe in the space available. It was common to be a polymath (multi-disciplined) 'back in the day'. We will focus on those areas of discovery and invention most relevant to electricity and electronics.

Note that many people independently invented the same thing; those who get the most credit were not necessarily the original discoverers. Also, many inventions represent the culmination of the work of many people. Some inventions are not necessarily the result of the labours of any specific individual but result from many contributions.

We have tried to be as comprehensive as possible, but there will be inventions or inventors we have not been able to include in the available space.

This series contains six parts. This first part, and the following three that will be coming over the next few months, will detail various individual inventors, usually with multiple inventions over a range of dates, organised by their birth dates. The last two parts will mostly cover inventions attributed to companies or other organisations, such as universities.

We have endeavoured to use multiple sources to find accurate dates; dates often vary between sources, sometimes significantly.

Here is our list of inventors by date of birth (up to 1804):

Thales of Miletus *c.624BCE-c.546BCE*
static electricity

Described the generation of static electricity by rubbing amber, which caused it to attract feathers and other light materials. He also observed that lodestone, a form of magnetite naturally magnetised by lightning, could attract iron.

Theophrastus *c.371BCE-c.287BCE*
pyroelectricity

Is said to have discovered pyroelectricity, the property of a material to temporarily become charged when heated and attract light materials like ash, similar to when amber is rubbed.

William Gilbert *1544-1603*
electricity

Coined the term "electricus", from which the word electricity is derived. He also explained that compasses worked because the Earth is a giant magnet with an iron core. He wrote a book in 1600 with the title "De Magnete". You can read that book at www.gutenberg.org/ebooks/33810

He also invented the instrument

Recent articles in Silicon Chip magazine on electronics tech.

All About Batteries, January–March 2022; siliconchip.au/Series/375
The History of Transistors, March–May 2022; siliconchip.au/Series/378
IC Fabrication, June & July 2022; siliconchip.au/Series/382
Display Technologies, September & October 2022; siliconchip.au/Series/387
Computer Memory, January & February 2023; siliconchip.au/Series/393

The History of Electronics, part one

now known as the electroscope, which detects the presence of electric charge. Gilbert mistakenly did not believe electricity and magnetism were related; Hans Christian Ørsted and James Clerk Maxwell later showed them to be.

Otto von Guericke *1602-1686*
electrostatic generator

He invented the first electrostatic generator, a sulfur sphere that could be rubbed to impart an electric charge to attract or repel objects. It inspired other, more advanced frictional generators.

Christiaan Huygens *1629-1695*
wave theory of light

Developed the wave theory of light in 1690, which related to electric and magnetic fields.

Francis Hauksbee the Elder *1660-1713*
modified electrostatic generator

Made a modified version of Otto von Guericke's electrostatic generator in 1705, a partially evacuated glass sphere into which mercury was introduced. If rubbed to generate a charge, a glow was produced where the glass was touched.

This led to the much later development of the gas discharge lamp, neon lighting and mercury vapour lamps. You can read his book "Physico-mechanical experiments" at https://catalogue.nla.gov.au/catalog/3171279

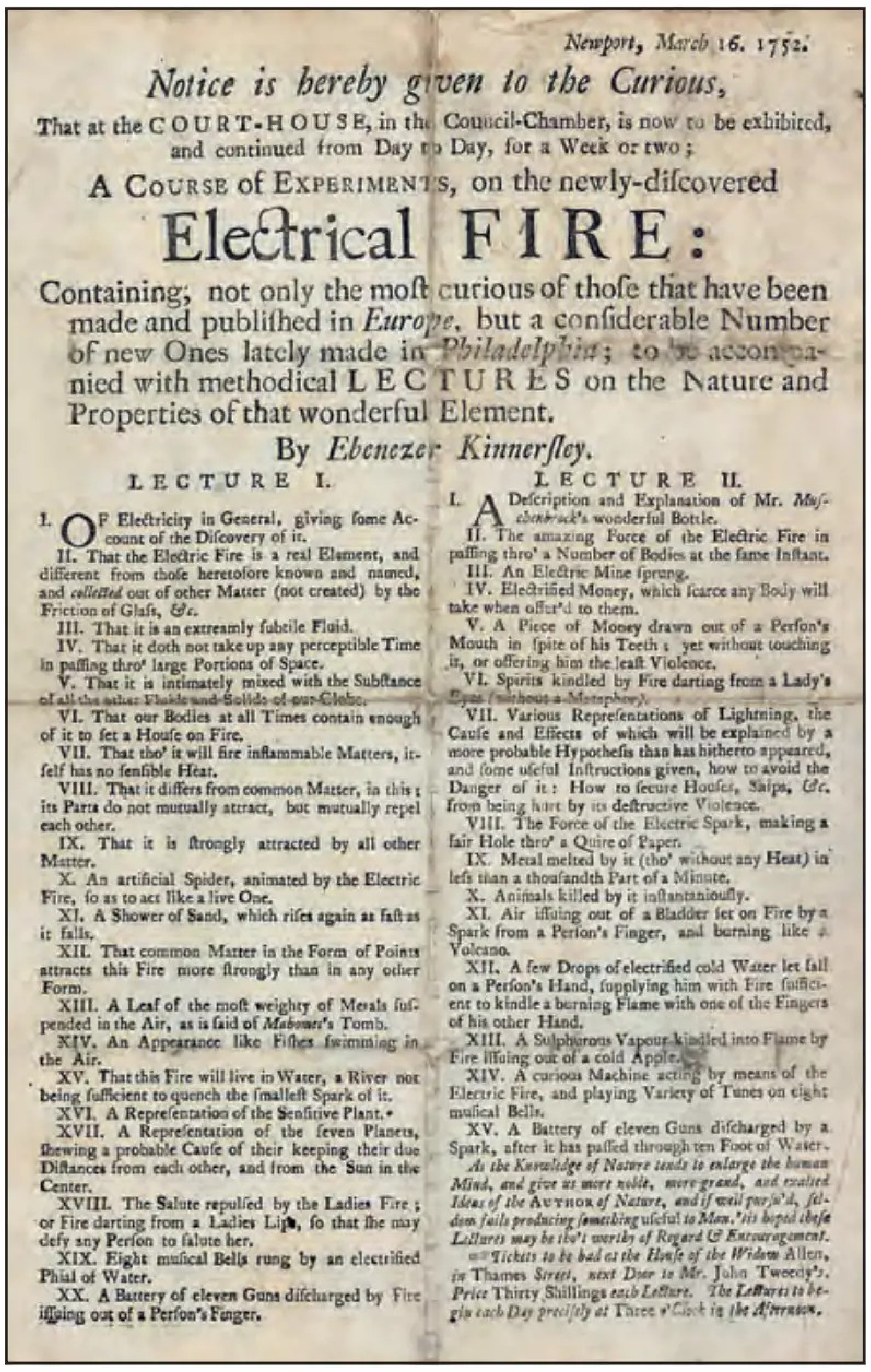
Johann George Schmidt *unknown*
pyroelectricity

Observed in 1707 that the mineral tourmaline had a property we now know as pyroelectricity.

Stephen Gray *1666-1736*
electrical conductivity

Discovered the principles of electrical conductivity and distinguished between conductors and insulators. He also made discoveries in electrical induction, imparting a charge into another object without contact. He received little credit for his discoveries, but today, some know him as the "father of electricity".

Christiaan Huygens also invented the pendulum clock. *Source: https://w.wiki/7ATc*



Newport, March 16. 1752.

Notice is hereby given to the Curious,

That at the COURT-HOUSE, in the Council-Chamber, is now to be exhibited, and continued from Day to Day, for a Week or two;

A COURSE of EXPERIMENTS, on the newly-diſcovered

Electrical FIRE:

Containing, not only the moſt curious of thoſe that have been made and publiſhed in *Europe*, but a conſiderable Number of new Ones lately made in *Philadelphia*; to be accompanied with methodical LECTURES on the Nature and Properties of that wonderful Element.

By *Ebenezer Kinnerſley.*

LECTURE I.

I. OF Electricity in General, giving ſome Account of the Diſcovery of it.

II. That the Electric Fire is a real Element, and different from thoſe heretofore known and named, and *collected* out of other Matter (not created) by the Friction of Glaſs, *&c.*

III. That it is an extreamly ſubtile Fluid.

IV. That it doth not take up any perceptible Time in paſſing thro' large Portions of Space.

V. That it is intimately mixed with the Subſtance of all the other Fluids and Solids of our Globe.

VI. That our Bodies at all Times contain enough of it to ſet a Houſe on Fire.

VII. That tho' it will fire inflammable Matters, itſelf has no ſenſible Heat.

VIII. That it differs from common Matter, in this; its Parts do not mutually attract, but mutually repel each other.

IX. That it is ſtrongly attracted by all other Matter.

X. An artificial Spider, animated by the Electric Fire, ſo as to act like a live One.

XI. A Shower of Sand, which riſes again as faſt as it falls.

XII. That common Matter in the Form of Points attracts this Fire more ſtrongly than in any other Form.

XIII. A Leaf of the moſt weighty of Metals ſuſpended in the Air, as is ſaid of *Mahomet's* Tomb.

XIV. An Appearance like Fiſhes ſwimming in the Air.

XV. That this Fire will live in Water, a River not being ſufficient to quench the ſmalleſt Spark of it.

XVI. A Repreſentation of the Senſitive Plant.

XVII. A Repreſentation of the ſeven Planets, ſhewing a probable Cauſe of their keeping their due Diſtances from each other, and from the Sun in the Center.

XVIII. The Salute repulſed by the Ladies Fire; or Fire darting from a Ladies Lips, ſo that ſhe may defy any Perſon to ſalute her.

XIX. Eight muſical Bells rung by an electrified Phial of Water.

XX. A Battery of eleven Guns diſcharged by Fire iſſuing out of a Perſon's Finger.

LECTURE II.

I. A Deſcription and Explanation of Mr. *Muſchenbrock's* wonderful Bottle.

II. The amazing Force of the Electric Fire in paſſing thro' a Number of Bodies at the ſame Inſtant.

III. An Electric Mine ſprung.

IV. Electrified Money, which ſcarce any Body will take when offer'd to them.

V. A Piece of Money drawn out of a Perſon's Mouth in ſpite of his Teeth; yet without touching it, or offering him the leaſt Violence.

VI. Spirits kindled by Fire darting from a Lady's Eyes (without a Metaphor).

VII. Various Repreſentations of Lightning, the Cauſe and Effects of which will be explained by a more probable Hypotheſis than has hitherto appeared, and ſome uſeful Inſtructions given, how to avoid the Danger of it: How to ſecure Houſes, Ships, *&c.* from being hurt by its deſtructive Violence.

VIII. The Force of the Electric Spark, making a fair Hole thro' a Quire of Paper.

IX. Metal melted by it (tho' without any Heat) in leſs than a thouſandth Part of a Minute.

X. Animals killed by it inſtantaniouſly.

XI. Air iſſuing out of a Bladder ſet on Fire by a Spark from a Perſon's Finger, and burning like a Volcano.

XII. A few Drops of electrified cold Water let fall on a Perſon's Hand, ſupplying him with Fire ſufficient to kindle a burning Flame with one of the Fingers of his other Hand.

XIII. A Sulphurous Vapour kindled into Flame by Fire iſſuing out of a cold Apple.

XIV. A curious Machine acting by means of the Electric Fire, and playing Variety of Tunes on eight muſical Bells.

XV. A Battery of eleven Guns diſcharged by a Spark, after it has paſſed through ten Foot of Water.

As the Knowledge of Nature tends to enlarge the human Mind, and give us more noble, more grand, and exalted Ideas of the AUTHOR *of Nature, and if well purſu'd, ſeldom fails producing ſomething* uſeful *to Man. 'tis hoped theſe Lectures may be tho't worthy of Regard & Encouragement.*

☞ *Tickets to be had at the Houſe of the Widow* Allen, *in* Thames *Street, next Door to Mr.* John Tweedy's. *Price* Thirty Shillings *each Lecture. The Lectures to begin each Day preciſely at* Three o'Clock *in the Afternoon.*

Fig.1: Ebenezer Kinnersley's 'Electrical FIRE' lecture notice. ***Source: Brown University Library (https://library.brown.edu/dps/curio/2013/05/).***

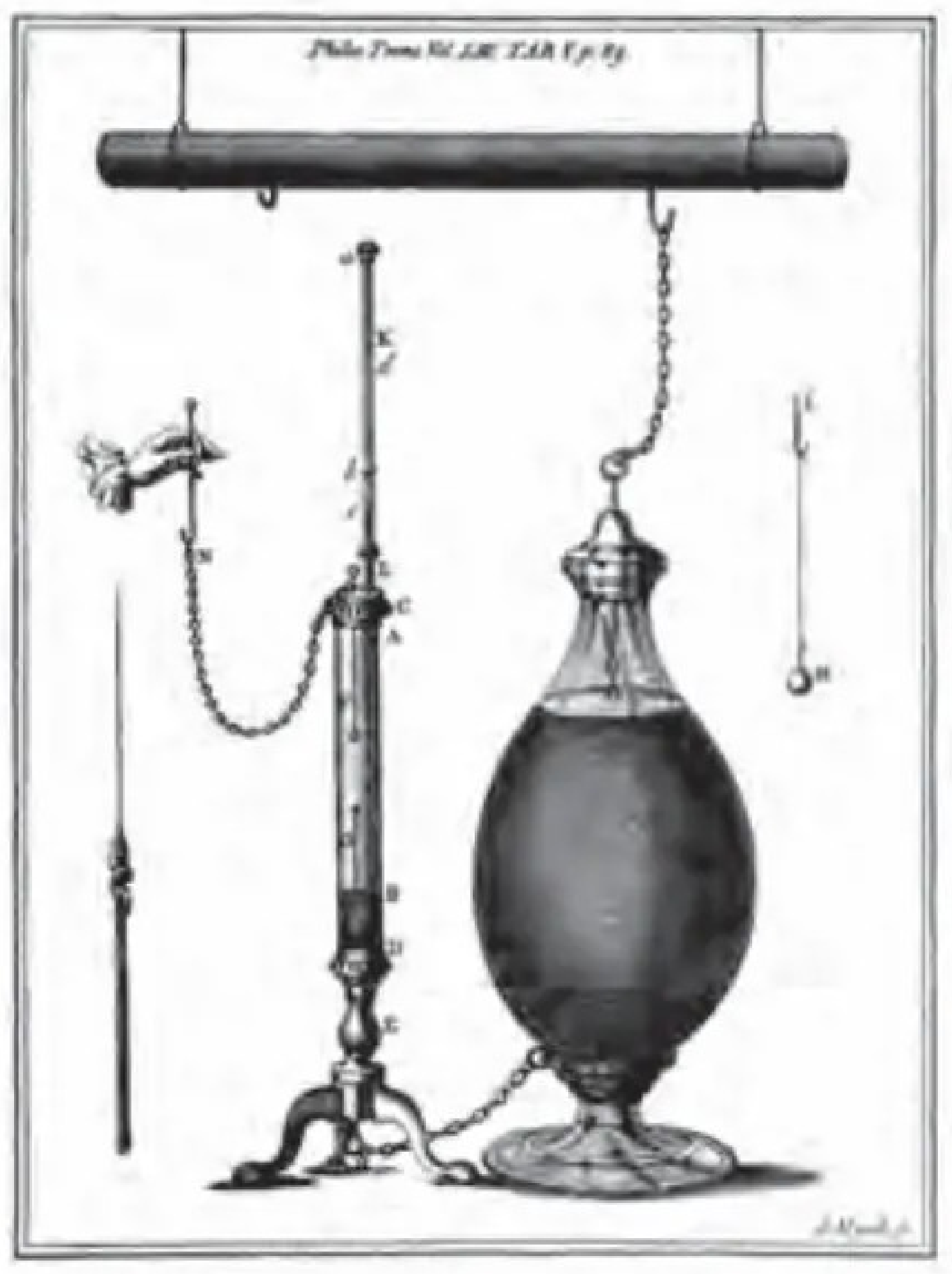


Fig.2: Kinnersley's "electric air thermometer" used a spark discharge to push water up a tube. ***Source: https://w.wiki/78sQ***

Pieter van Musschenbroek *1692-1761*
Leyden jar / capacitor

Along with his student and a collaborator, he invented what became known as the Leyden jar in 1756, the original capacitor. It was used to store electrical energy produced by frictional generators. It consisted of a glass jar filled with water, a brass rod and another conductor.

You can easily make a Leyden jar; see the video from ElectroBOOM at https://youtu.be/xjW-isgOijs and www.wikihow.com/Make-a-Leyden-Jar

Ewald Georg von Kleist *1700-1748*
Kleistian jar

Invented the Kleistian jar in 1745, a form of Leyden jar.

Benjamin Franklin *1706-1790*
lightning rods, glass harmonica etc

He named positive and negative charges in 1747. In 1748, he constructed a multi-plate capacitor with glass and lead plates. In that same year, he invented the "electric wheel", a type of electrostatic motor that would run at 12-15RPM from a charge supplied by Leyden jars.

In 1750, he showed that Leyden jars discharged more easily near a pointed rod, leading to the invention of lightning rods (also see Kinnersley's entry). In 1752, he flew a kite in a thunderstorm to charge a Leyden jar attached to the wet string, proving that lightning was electricity. (The following two people who tried that were electrocuted.)

In 1751, he published a series of pamphlets on electricity (pemag.au/link/abnr).

Carl Linnaeus *1707-1778*
pyroelectricity

Determined that pyroelectricity was a type of electricity in 1747. He later became known as Carl von Linné after being ennobled

Ebenezer Kinnersley *1711-1778*
electric fire / electricity

Performed experiments with "electric fire", as electricity was then called – see Fig.1.

Benjamin Franklin described him as "an ingenious neighbor". In 1748, he discovered that electricity passed through water. In 1751-2, he held a series of lectures about electric fire.

In his March 1752 lecture, he suggested the lightning rod to protect structures from lightning before

Benjamin Franklin was one of the Founding Fathers of the USA. *Source: https://w.wiki/7ATw*

Franklin did his kite experiment. In 1761, he wrote a letter to Franklin and, in 1763, published details of an "electric air thermometer" – see Fig.2.

He demonstrated that electricity could produce heat. In 1761, he used electricity from Leyden jars to heat metals to incandescence, producing visible light, paving the way for the light globe. See "Expt. 11" in his letter to Franklin (**pemag.au/link/abp2**).

You can find instructions to make a "proof of concept" light globe on Hackaday: **pemag.au/link/abnk**

Andrew Gordon 1712-1751
electrostatic reaction motor

Invented the "electrical whirl" (Fig.3), described in 1745 (**pemag.au/link/abnt**). It was an electrostatic reaction motor, also demonstrated by Benjamin Franklin in 1748. It was incapable of useful work, but some call it the first electric motor.

It consisted of a free-spinning star with angled, pointed ends that were charged from a power source. Ionised gas from the tips caused it to rotate. For more details, see **pemag.au/link/abn2**

Franz Aepinus 1724-1802
electricity and magnetism

He was the first to publish a treatise on electricity and magnetism (see his book at: **pemag.au/link/abnu**).

Johan Carl Wilcke 1732-1796
electrophorus

Invented the electrophorus, a device to produce static electricity, in 1762.

Luigi Galvani 1737-1798
bio-electricity

He is famous for discovering that frog's legs will twitch with the application of an electric discharge from a charged Leyden jar. He also made the legs move with two differing metals that generated a current like a battery. Due to this early work in the field of bioelectricity, many modern electrical-related phenomena are named after Galvani.

Charles Coulomb 1737-1806
Coulomb's law / electric charge

Invented the torsion balance, which enabled him to measure forces of attraction or repulsion between charged or magnetised bodies. **Coulomb's law** states that the force between two electrically charged bodies is proportional to the magnitude of their electric charge and the inverse square of the distance between them.

This was known earlier, but it is named after Coulomb, as he was the first to publish it in 1785. The **Coulomb (C)** is also the unit of electric charge.

Alessandro Volta 1745-1827
battery (voltaic pile)

He improved the electrophorus in 1775. Then in 1800, he invented what is now known as the voltaic pile or battery made of copper and zinc, using either saltwater or sulfuric acid electrolyte. He acknowledged the contributions of William Nicholson, Tiberius Cavallo and Abraham Bennet to his battery work.

The unit of electrical potential, the **volt (V)**, was named in his honour. He discovered by accident that a short circuit of his voltaic pile caused a copper wire to glow, confirming the principle of the incandescent light globe.

Pierre-Simon Laplace 1749-1827
Laplace transform

Developed the **Laplace transform** in 1785. It is used to solve differential equations, making it essential for circuit analysis.

Vasily Vladimirovich Petrov 1761-1834
electric arc – welding

Discovered the electric arc in 1802 after he built the world's largest voltaic pile, comprising 4200 copper and zinc discs. In 1803, he proposed several uses for the electric arc, such as lighting, welding, metal processing etc.

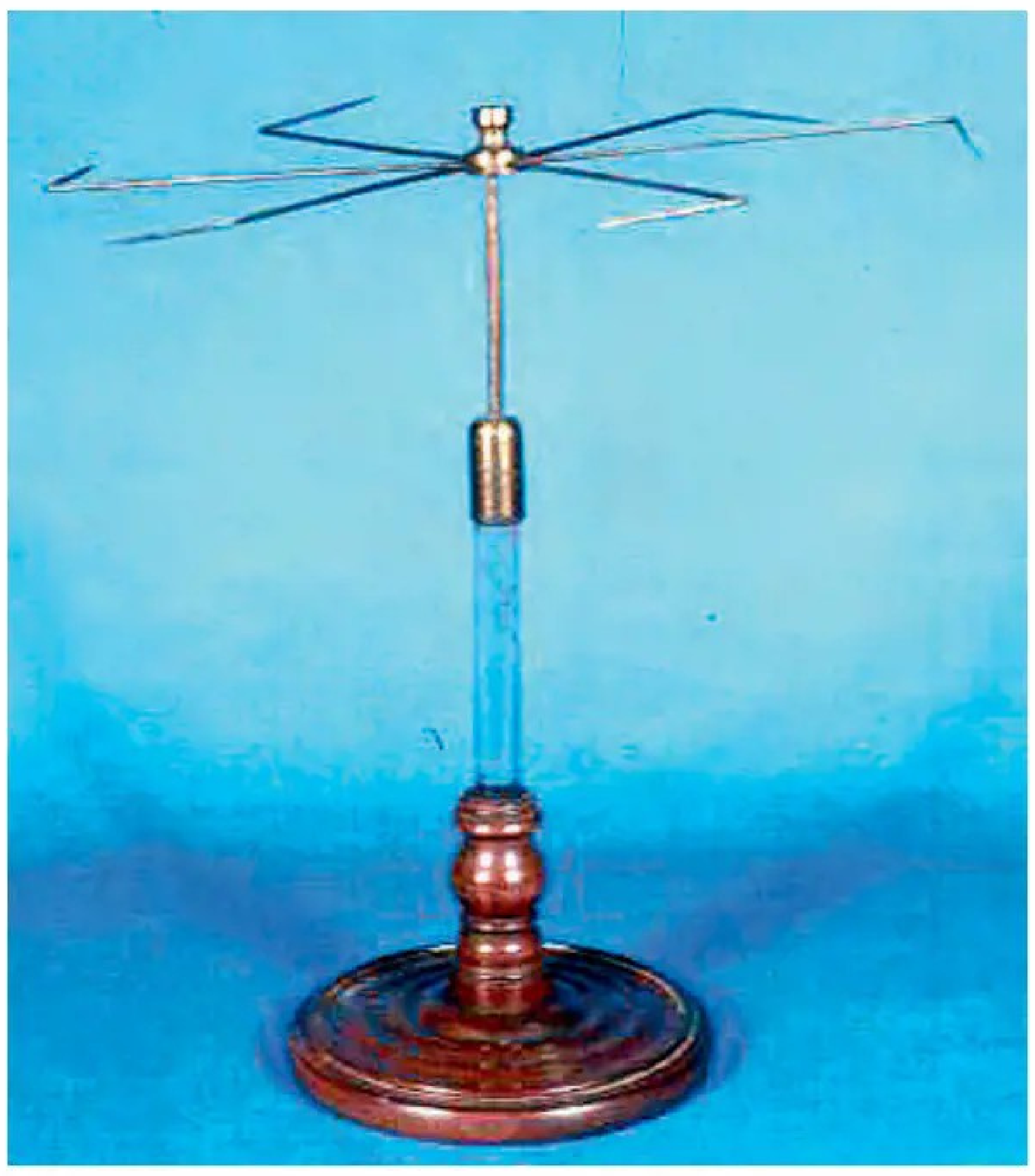


Fig.3: an electric whirl similar to the one invented by Andrew Gordon in 1745. This one is on display in the physics department of Washington and Lee University. *Source: http://physics.kenyon.edu/EarlyApparatus/Static_Electricity/Electric_Whirl/Electric_Whirl.html*

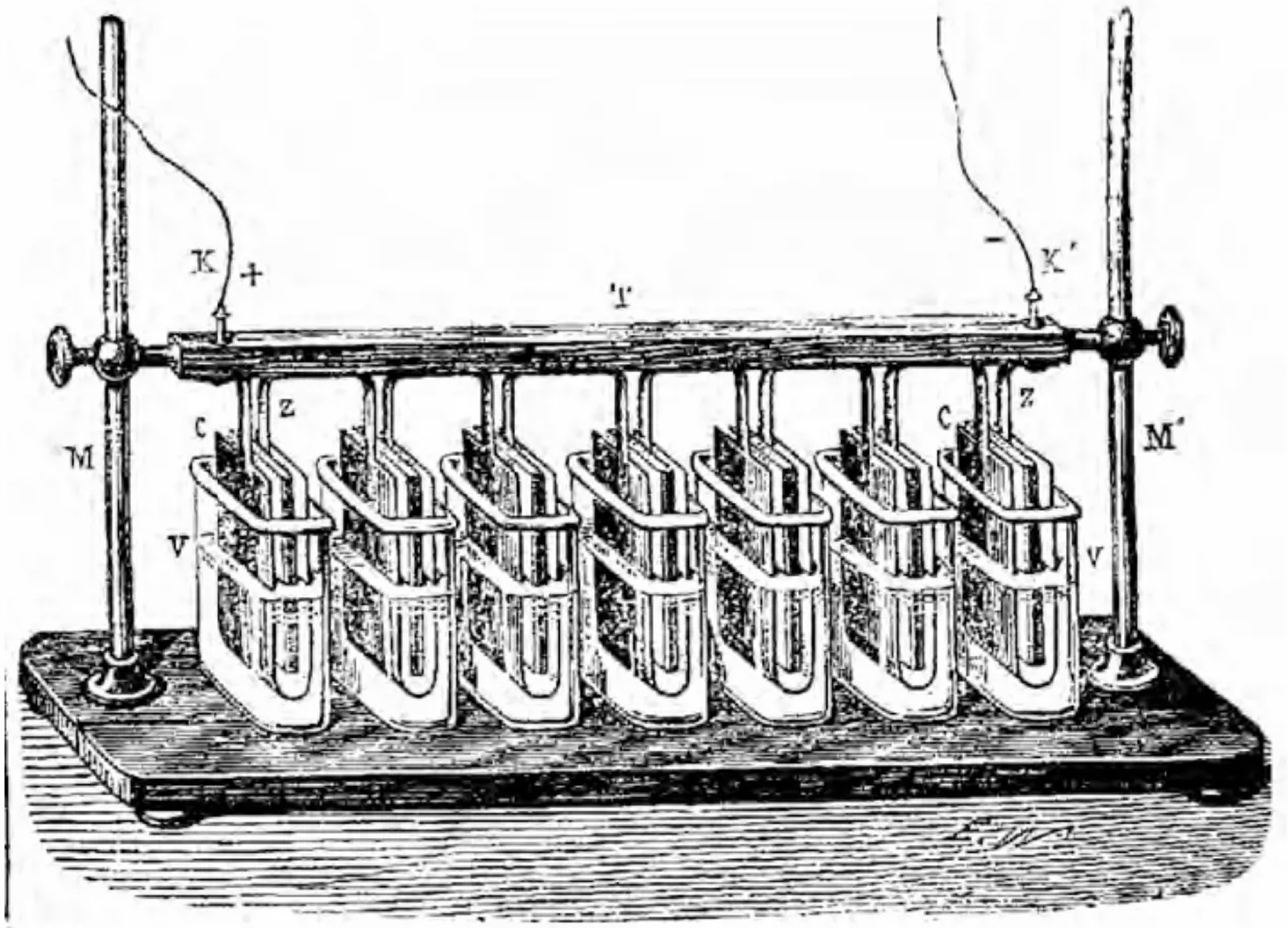


Fig.4: Wollaston's improved battery with removable electrodes. *Source: https://w.wiki/78sR*

The History of Electronics, part one

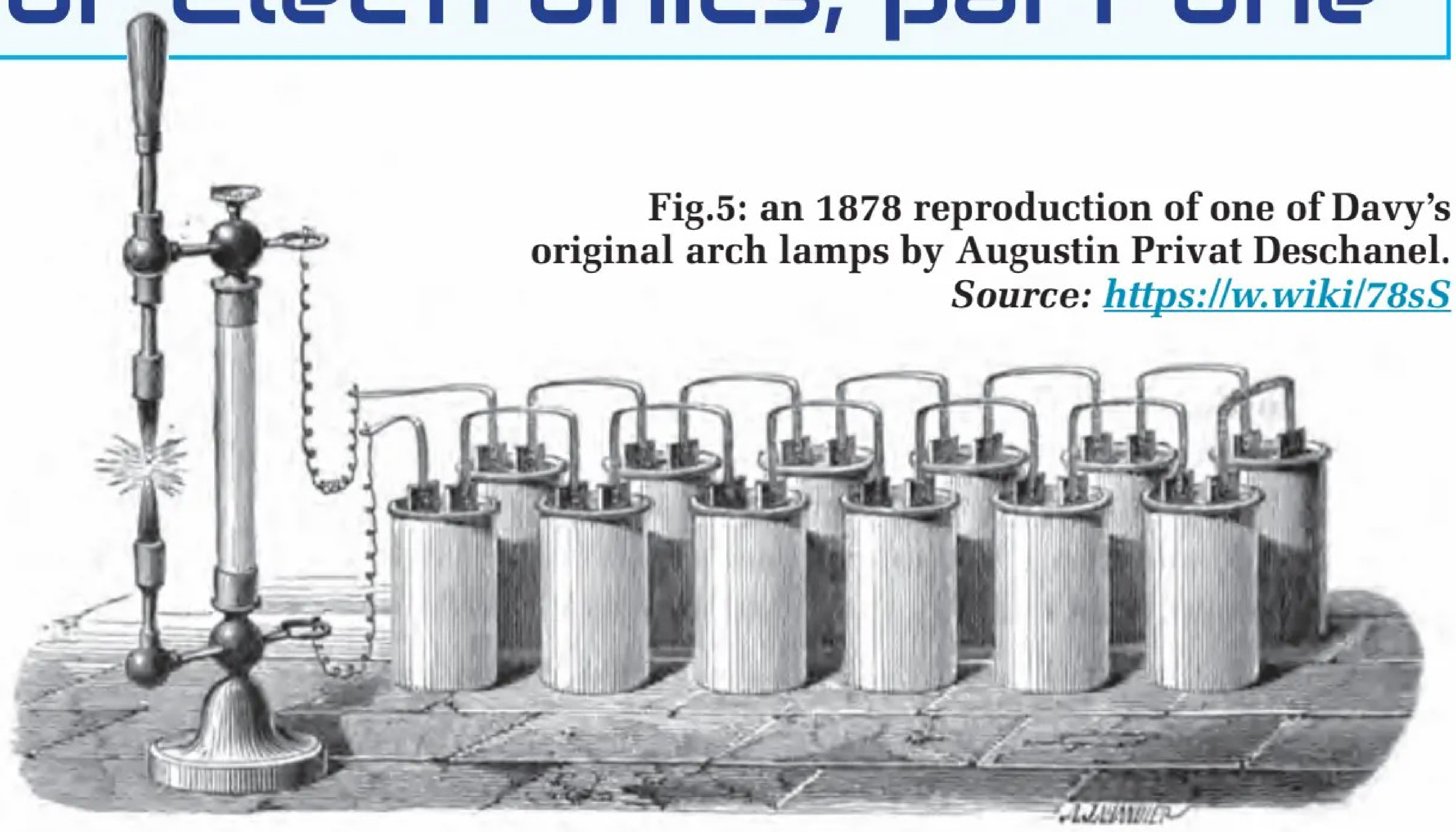


Fig.5: an 1878 reproduction of one of Davy's original arch lamps by Augustin Privat Deschanel. *Source: https://w.wiki/78sS*

William Hyde Wollaston *1766-1828*

static electricity and electromagnetic induction

Demonstrated that static electricity was the same as from voltaic piles in 1801. He was said to have "accidentally" discovered electromagnetic induction 10 years before Faraday (who made the discovery in 1831) and made a failed attempt to build an electric motor.

He built an improved type of copper/zinc battery in which the electrodes were raised from the electrolyte when not in use, improving the life – see Fig.4.

John Dalton *1766-1844*

atomic theory – materials

Contributed to atomic theory in ways that improved the understanding of conductors, insulators and semiconductors.

Thomas Johann Seebeck *1770-1831*

thermocouples / thermopiles

Discovered in 1822 that a junction of two dissimilar metals produced a current. This is the basis of thermocouples, used to measure temperature, and thermopiles, which convert heat into electricity (such as radioisotope thermoelectric generators on spacecraft).

Thomas Young *1773-1829*

expanded on wave theory

He expanded on the wave theory of light (first described by Huygens), vision and colour theory.

André-Marie Ampère *1775-1836*

Ampère's force law and solenoid

Set out to discover the relationship between electricity and magnetism. In 1820, Ampère's friend, Dominique François Jean Arago, demonstrated the discovery of Hans Christian Ørsted that a current-carrying wire deflects a magnetised needle.

Ampère determined that two parallel current-carrying wires would either attract or repel each other depending on the relative current directions and established **Ampère's force law**. He invented the solenoid and had an idea for an electric telegraph. The SI unit for electric current, the **amp (A)**, is named after him. Inspired by Ørsted, he also established **Ampère's right-hand grip rule**.

Carl Friedrich Gauss *1777-1855*

ionosphere and electric telegraph

Popularised **Gauss' law** in 1813, although it had already been discovered by Joseph Louis Lagrange in 1762.

In 1839, he postulated that an electrically conducting region of the atmosphere, now known as the ionosphere, reflected radio waves. The unit of magnetic induction, the **gauss (G)**, is named after him. He had achievements in many other areas. He worked with Wilhelm Eduard Weber to develop an electric telegraph in 1833.

Hans Christian Ørsted *1777-1851*

Oersted's law and right-hand thumb rule

Discovered in 1820 that the needle of a compass would deflect near a current-carrying wire, establishing that an electric current had a magnetic field, the first connection between electricity and magnetism. He established **Oersted's (or Ørsted's) law** which states that an electric current establishes a magnetic field around it.

That led to the "**right-hand thumb rule**", which describes the relationship between a current and its magnetic field. A unit of magnetic field strength, the **oersted (Oe)**, is named after him.

Sir Humphry Davy 1st Baronet *1778-1829*

carbon arch lamp

Invented the carbon arch lamp, later renamed from arch to arc (see Fig.5), in 1802, 1805, 1807 or 1809 (depending on the source). He used charcoal sticks and a 2000-cell battery to strike an arc across a 100mm gap. The electrodes were originally horizontal, and the arc was shaped like an arch, hence the name.

Arc lamps were widely used for street and commercial lighting from the 1870s until they were replaced by incandescent lighting from the early 1900s (except for specific applications like searchlights and movie projectors).

Movie reels used to commonly be 2000ft (610m) long, with a runtime of about 22 minutes. That coincided with the life of carbon rods in pre-1970s theatre projectors. The projectionist would change the carbon rods at the same time as the reel.

In 1801 or 1802, Davy also connected a piece of platinum across a 2000-cell battery, which caused it to glow, the basis for later experiments in incandescent lighting.

Michael Faraday was Davy's assistant from 1813 to about 1815, and occasionally helped him after that, such as with the Miner's Safety Lamp.

William Sturgeon *1783-1850*

electromagnet

Invented the electromagnet in 1824 – see Fig.6. It comprised 18 turns of copper wire on a lacquered iron U-shaped core, 30cm long and with a 13mm diameter. It was powered by a copper-zinc-acid battery. The cups contain mercury to make electrical connections. The magnet could support 4kg.

Samuel Hunter Christie *1784-1865*

"diamond method" (Wheatstone Bridge)

Published the "diamond method" to compare resistances in 1833, a forerunner of the Wheatstone Bridge.

Baron Pavel Schilling *1786-1837*

Schilling telegraph

Made numerous contributions to telegraphy and other areas. One of those inventions was the Schilling

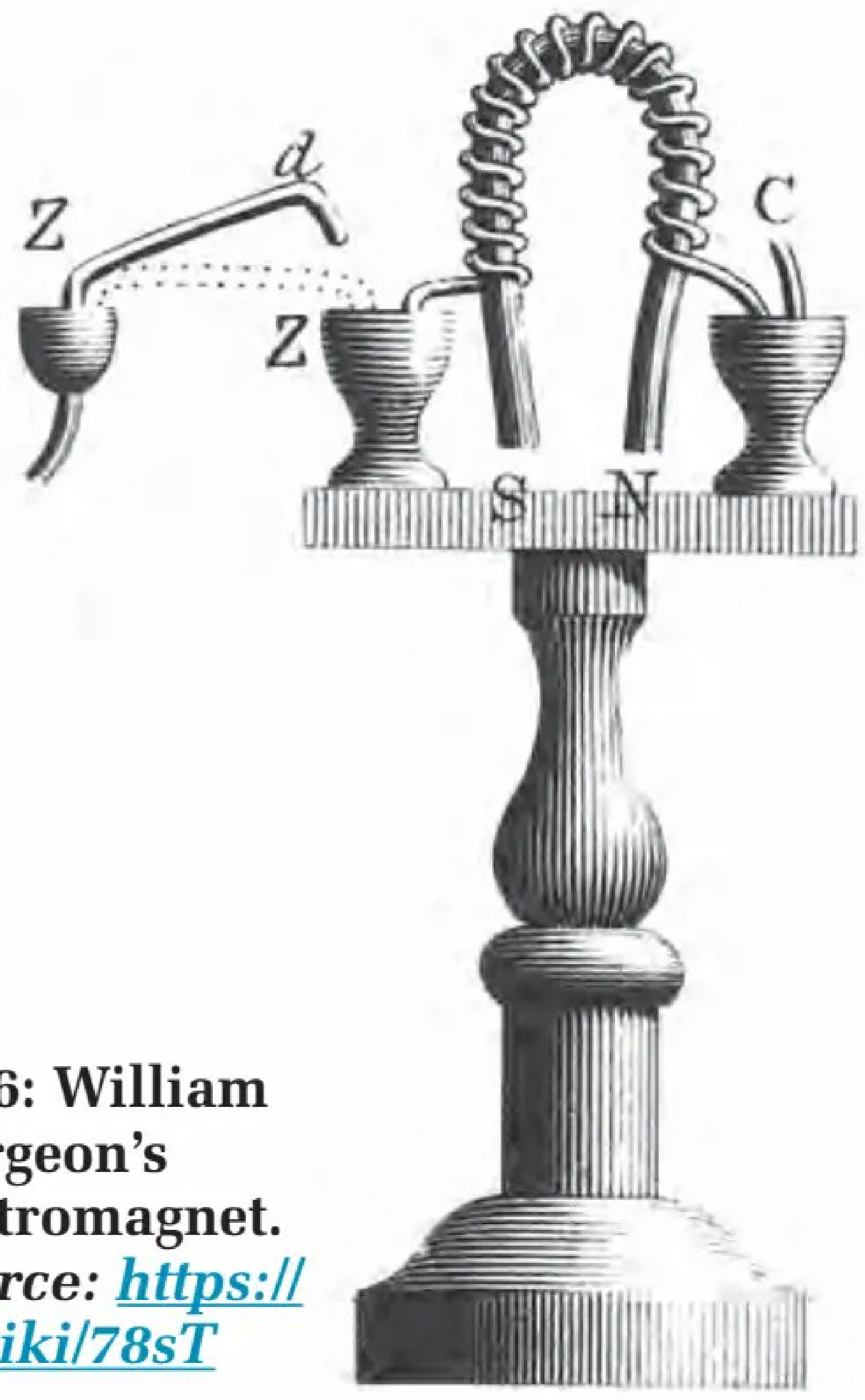


Fig.6: William Sturgeon's electromagnet. *Source: https://w.wiki/78sT*

telegraph, a type of 'needle telegraph' that sent a code along a series of wires to indicate the letter according to a binary code.

His first telegraph was shown in 1828. It used only two wires with an innovative variable-length binary code to encode 40 letters. The current direction also varied, so two wires could give eight different states. He demonstrated another instrument with six wires in 1832.

To transmit 40 different characters, six wires were needed for signalling, one for calling and one for a return. He abandoned the project because, from 1825, Czar Nicholas I of Russia opposed any form of mass communication and prohibited the public discussion of telegraphy.

Dominique François Jean Arago *1786-1853*
eddy currents

Conducted experiments with magnetism, mostly in 1823-1826. In 1824, he observed "rotary currents" or eddy currents. "Arago's rotations" demonstrated interactions between a spinning non-magnetic conductor such as a copper disc and a magnetised body like a compass needle or magnet.

Sir Francis Ronalds *1788-1873*
electric telegraph

Produced the first working electric telegraph in 1816. It was not until two decades later that commercialisation happened.

Georg Simon Ohm *1789-1854*
Ohm's law

Published "The Galvanic Circuit Investigated Mathematically" in 1827 – see pemag.au/link/abp3

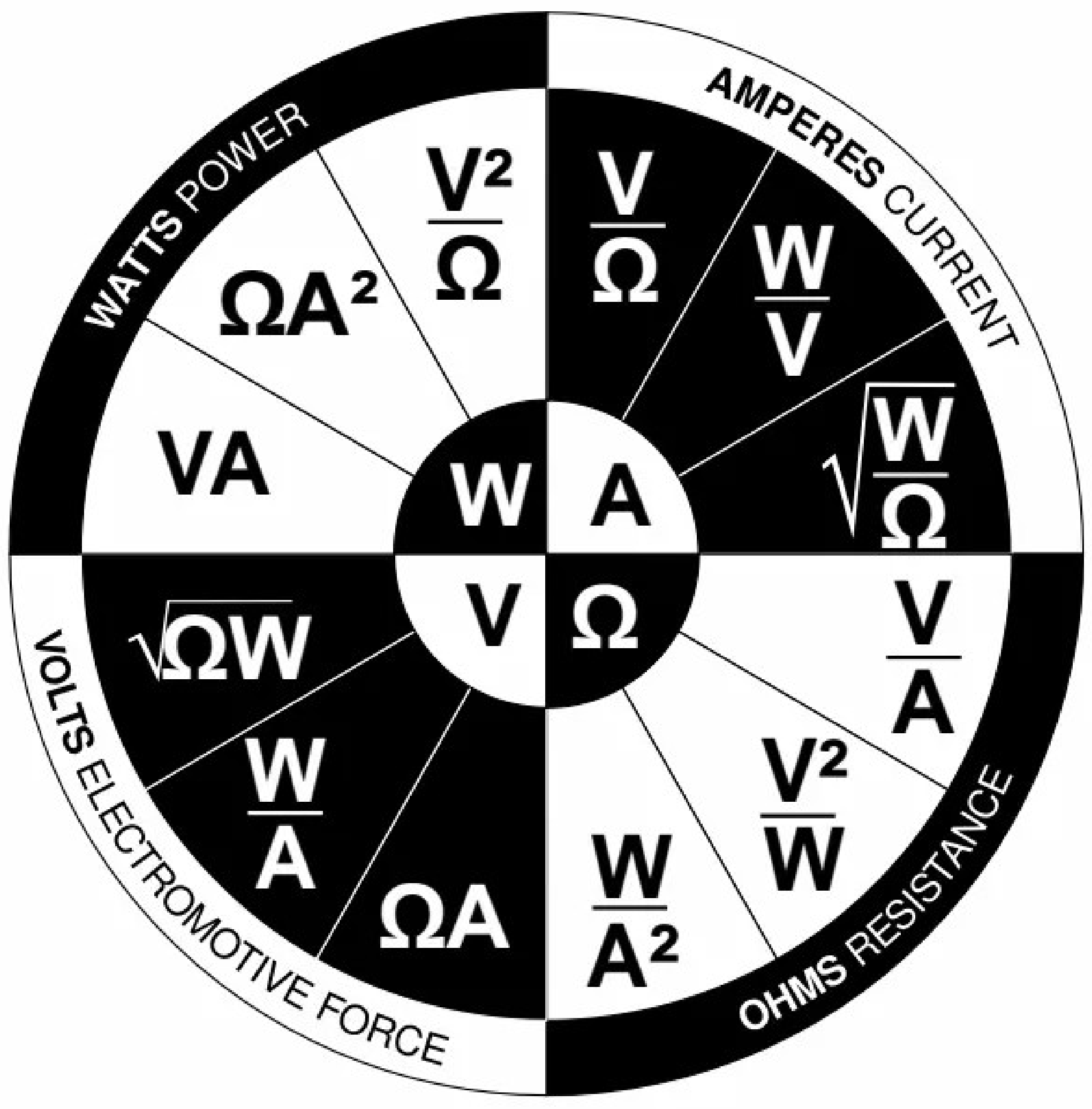


Fig.7: an Ohm's Law wheel calculator. *Source: https://w.wiki/78sV (CC-SA-3.0).*

In it, he detailed his theory of electricity, including the concept of resistance and what is now known as **Ohm's law** – see Fig.7. In 1825, he used different lengths of wire (10cm, 41cm, 183cm, 315cm and 762cm) to produce different resistances, deriving Ohm's law. It might be argued that he invented the resistor, although the concept of resistance was already known at the time. The unit **ohm (Ω)** is named after him.

Michael Faraday *1791-1867*
electromagnetic induction

Built a device to produce continuous "electromagnetic rotation", now called the homopolar motor (Figs.8 & 9) in 1821, soon after Ørsted discovered electromagnetism.

Faraday had discussed such a device with Sir Humphry Davy and William Hyde Wollaston, but failed to acknowledge them as contributing to his invention, causing controversy. See: pemag.au/link/abn4

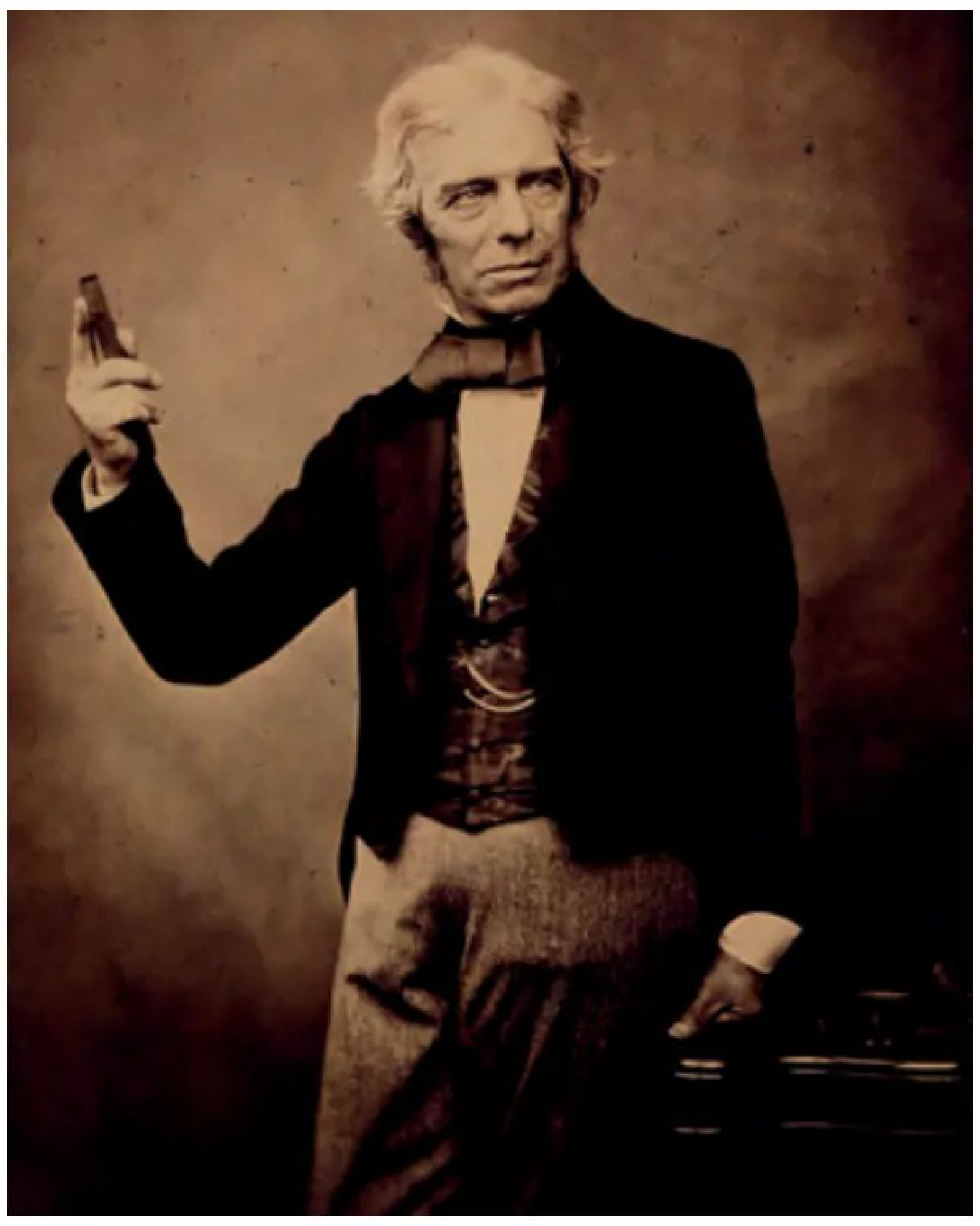
In 1831, Faraday discovered electromagnetic induction, demonstrating that a change in the magnetic field within a circuit induces an electromotive force (EMF) – see Fig.10. This discovery is the basis for electric power generation and led to the invention of the electrical generator and transformer. Joseph Henry independently discovered it in 1832, but Faraday published it first.

In 1833, he published "Faraday's laws of electrolysis", introducing terms such as electrode, anode, cathode, electrolyte and ion. He observed that the resistance of silver sulfide decreased as its temperature increased, the first mention of what we now call a thermistor, a semiconductor with a strongly temperature-dependent resistance. This was also the first observation of a semiconductor.

The unit of capacitance, the **farad (F)**, is named after him. Faraday also made numerous contributions in other areas; his theoretical work on the nature of the electromagnetic field led to the development of field theory in physics.

Samuel Morse *1791-1872*
Morse Code

Developed the concept of the single-wire telegraph and invented Morse Code in 1840 (later enhanced by Alfred Lewis Vail). In developing the telegraph, Morse had a problem of limited range, which he solved with the help of Professor Leonard Gale, by adding relay circuits.



Michael Faraday holding what is most likely ferromagnetic material. *Source: https://w.wiki/7AUi*

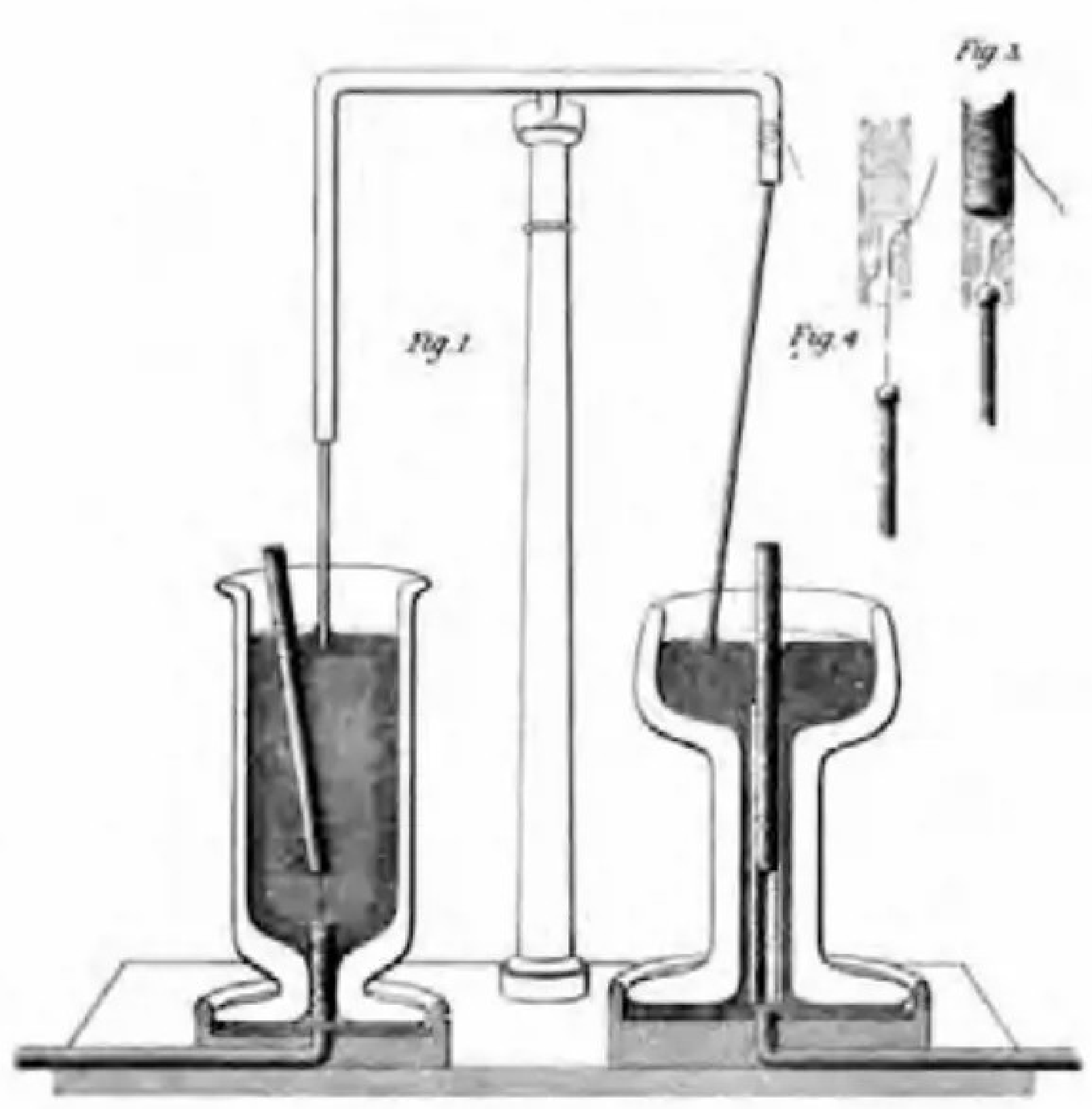


Fig.8: two versions of a magnetic rotation apparatus, the first motor. On the left, the lower magnetic rod rotates about the centre, while on the the right, the upper wire rotates about the centre magnet. The liquid is mercury. *Source: Michael Faraday.*

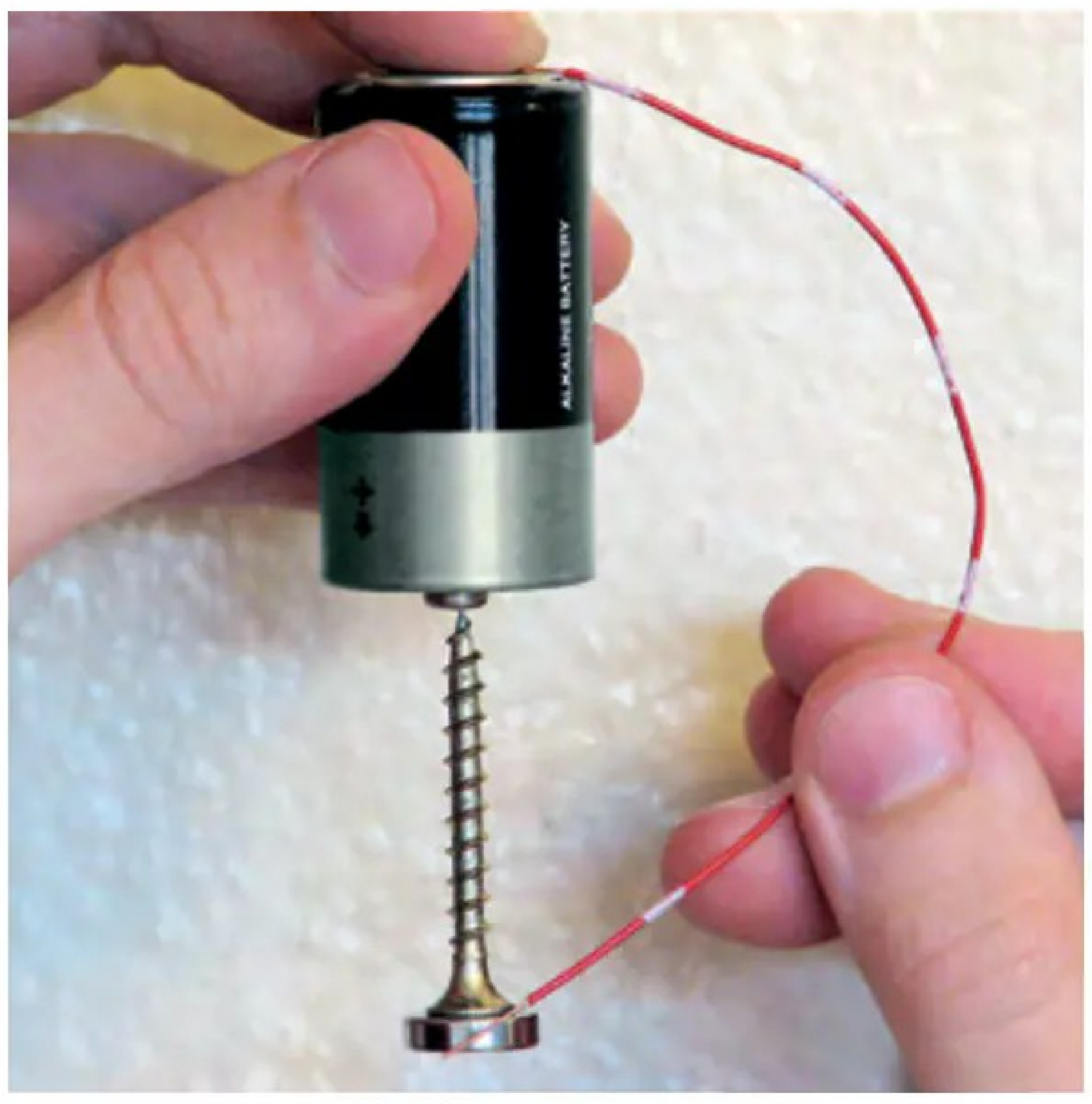


Fig.9: a simple homopolar motor you can make with a battery, a length of wire, a neodymium magnet and a steel screw. *Source: https://w.wiki/78sX (CC-BY-SA-2.5).*

The History of Electronics, part one

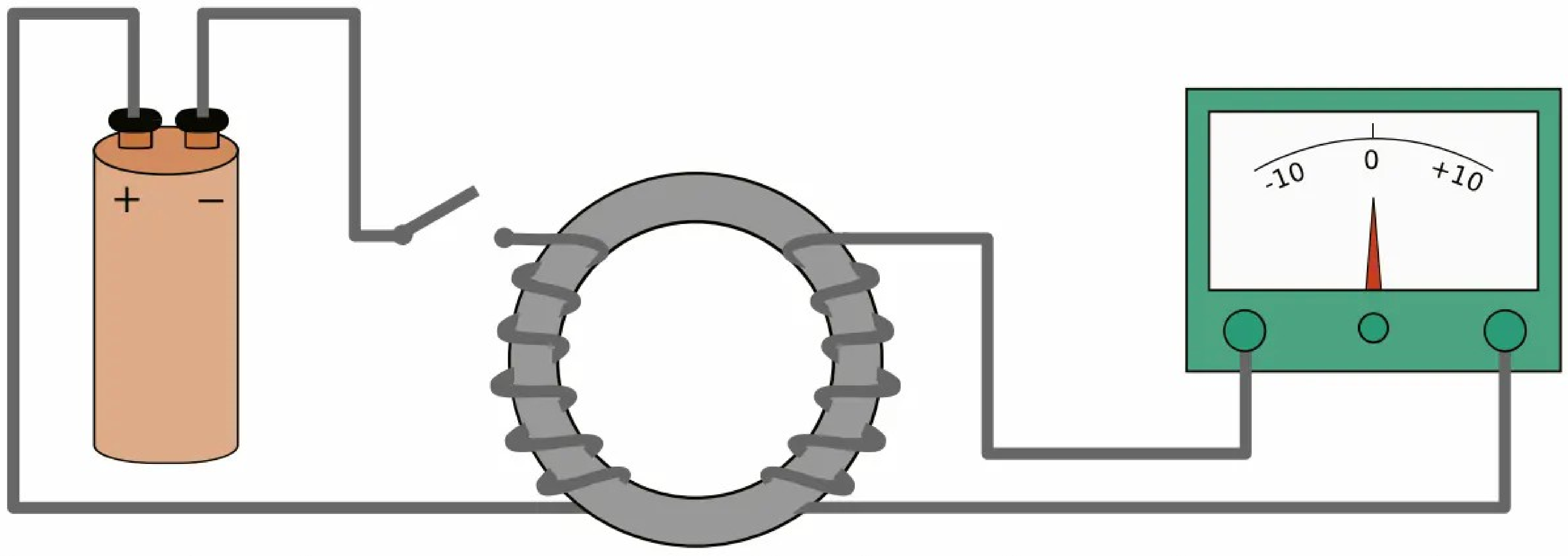


Fig.10: an iron ring apparatus used by Faraday to observe electromagnetic induction. Momentarily completing the circuit on the left resulted in a momentary current on the right. *Source:* https://w.wiki/78sW

Morse was contracted to build a 61km telegraph line between Washington, DC and Baltimore in 1843, which opened in 1844, with the first words transmitted being "What hath God wrought". By 1850, 19,300km of telegraph lines had been laid across the USA. Morse's 1840 telegraph patent can be seen at pemag.au/link/abn6

The Morse Code standard today (still in use by some radio hams) is defined by ITU-R M.1677-1 and is based upon the work of Friedrich Gerke in 1848, which led to the International Morse Code of 1865.

Marcellin Jobard 1792-1861
incandescent lighting

Suggested incandescent lighting in 1838, quoting É.M. Alglave and J. Boulard, "a small strip of carbon in a vacuum used as a conductor of a current, would emit an intense, fixed, and durable light". His student, Charles-François de Changy, commenced work on the idea in 1844.

Floris Nollet 1794-1853
magneto generator

Patented a magneto generator in 1850 for decomposing water by electrolysis. The hydrogen and oxygen produced were used in a form of stage lighting called limelight. The generator was also used for electric arc lighting and galvanising. The AC generated by the machine was converted to DC by a commutator.

Johann Poggendorff 1796-1877
slide wire potentiometer

Invented the slide wire potentiometer (variable resistor) in 1841. Around 1870, he also developed an electrostatic motor.

Joseph Henry 1799-1878
electromagnet and mutual inductance

Improved upon Sturgeon's electromagnet of 1824, in 1827, by using tightly wrapped silk-insulated wire rather than the uninsulated wire of Sturgeon – see Fig.11. This allowed Henry to use many layers of wire to make a more powerful magnet. He also discovered self-induction and mutual inductance.

In 1831, he made the world's first commercial electrical product, a powerful electromagnet to separate magnetite from crushed ore (see the video at https://youtu.be/ru-daEOuUjs). Also in 1831, he developed the first electric motor based on a rocking rather than rotary motion (see Fig.12).

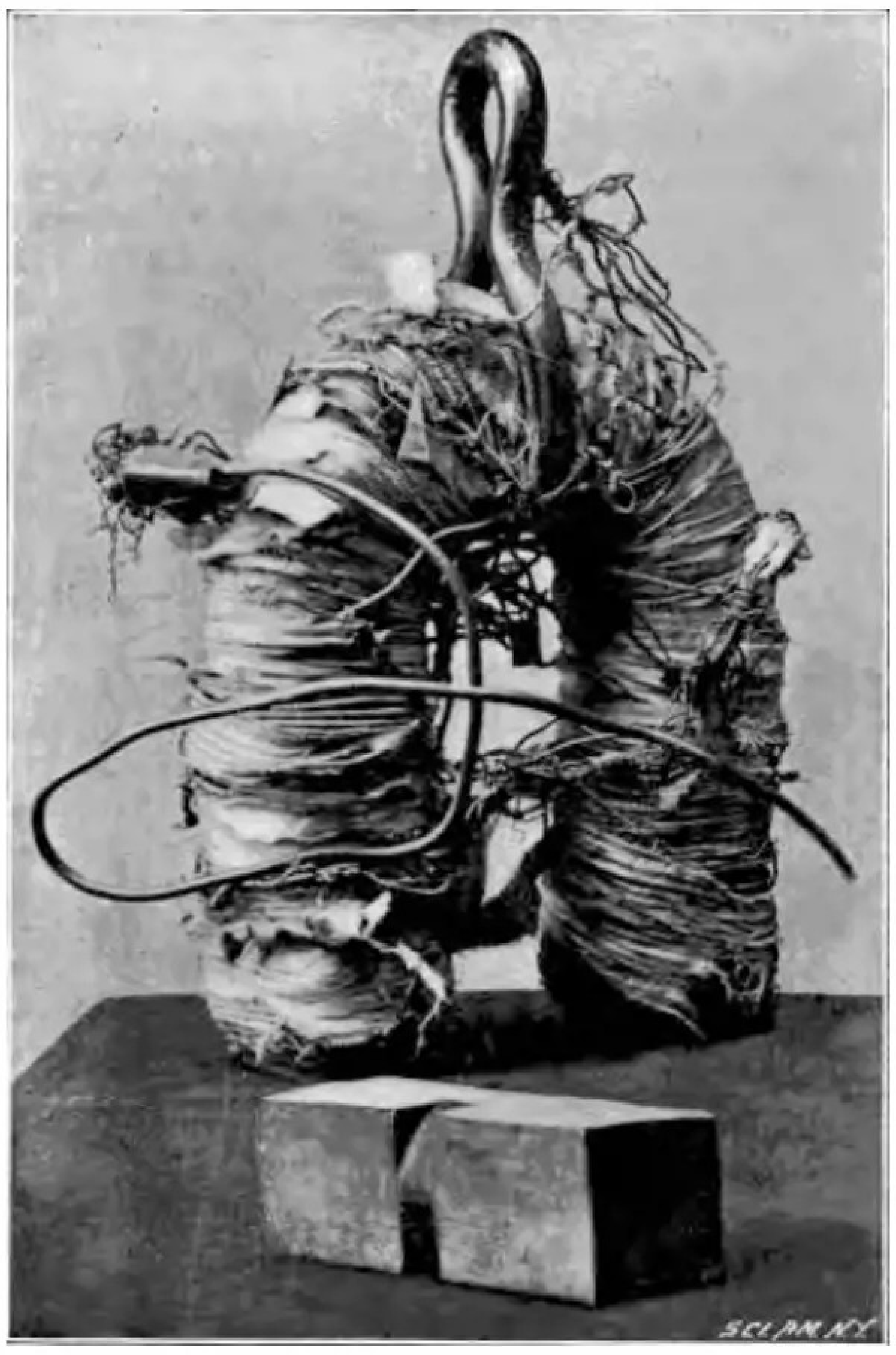


Fig.11: Joseph Henry's "intensity magnet". *Source:* https://w.wiki/78sY

The unit of inductance, the **henry (H)**, is named after him; it is thought that Henry discovered inductance before Faraday, but Faraday published his findings first.

Nicholas Joseph Callan 1799-1864
induction coil and Maynooth battery

He invented the induction coil in 1836. It is a form of transformer driven by a pulsating direct current at about 20Hz using an "interrupter" to make and break the current flow. Despite not inventing it, Heinrich Daniel Ruhmkorff patented it in 1851 and then commercialised it.

In 1848, he also commercialised the world's largest battery at the time, the

Joseph Henry in 1879. *Source:* https://w.wiki/7AU$

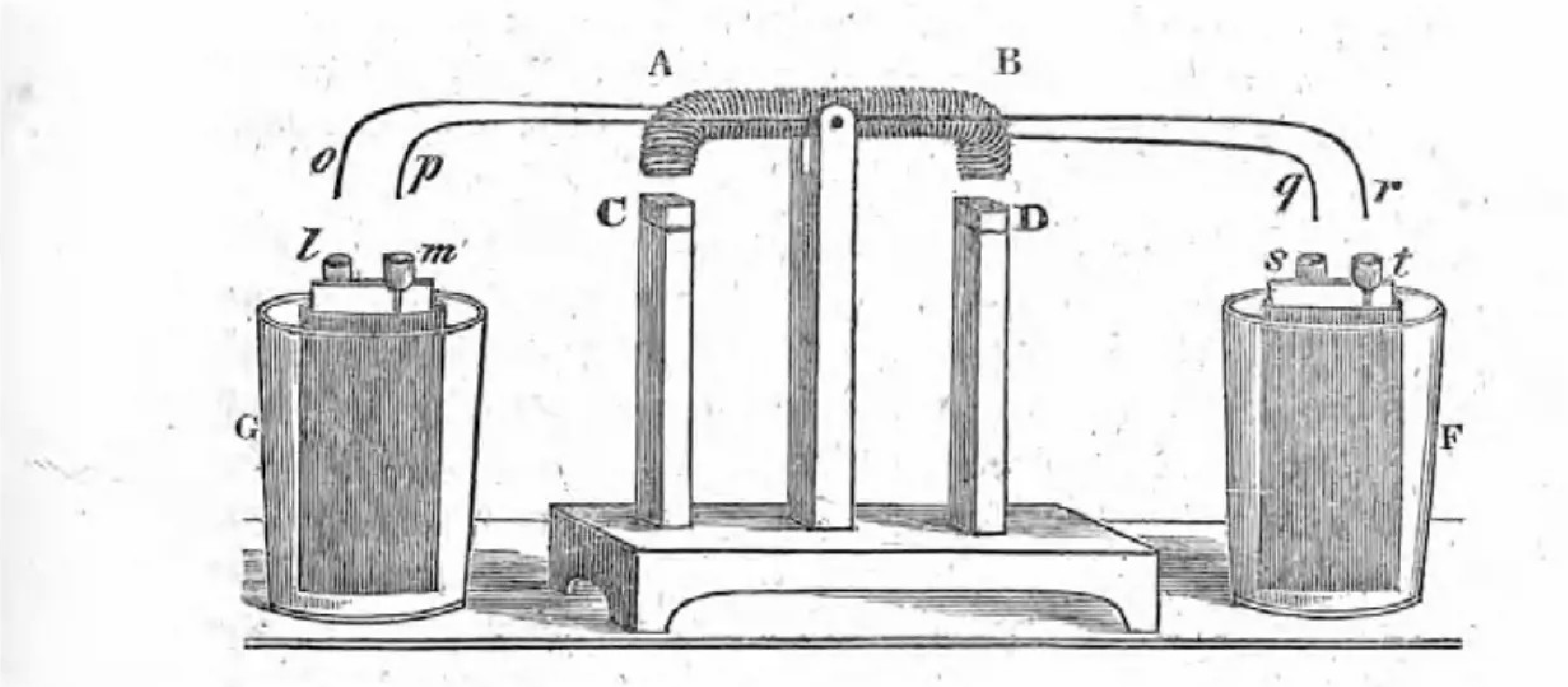


Fig.12: Joseph Henry's rocking beam electric motor of 1831. It pivoted in the middle with its ends in line with permanent magnets (C and D). As it rocked, electrodes contacted batteries at the ends (G and F), the magnet polarity reversed, and the beam would rock the other way. *Source:* https://siarchives.si.edu/collections/siris_sic_13161

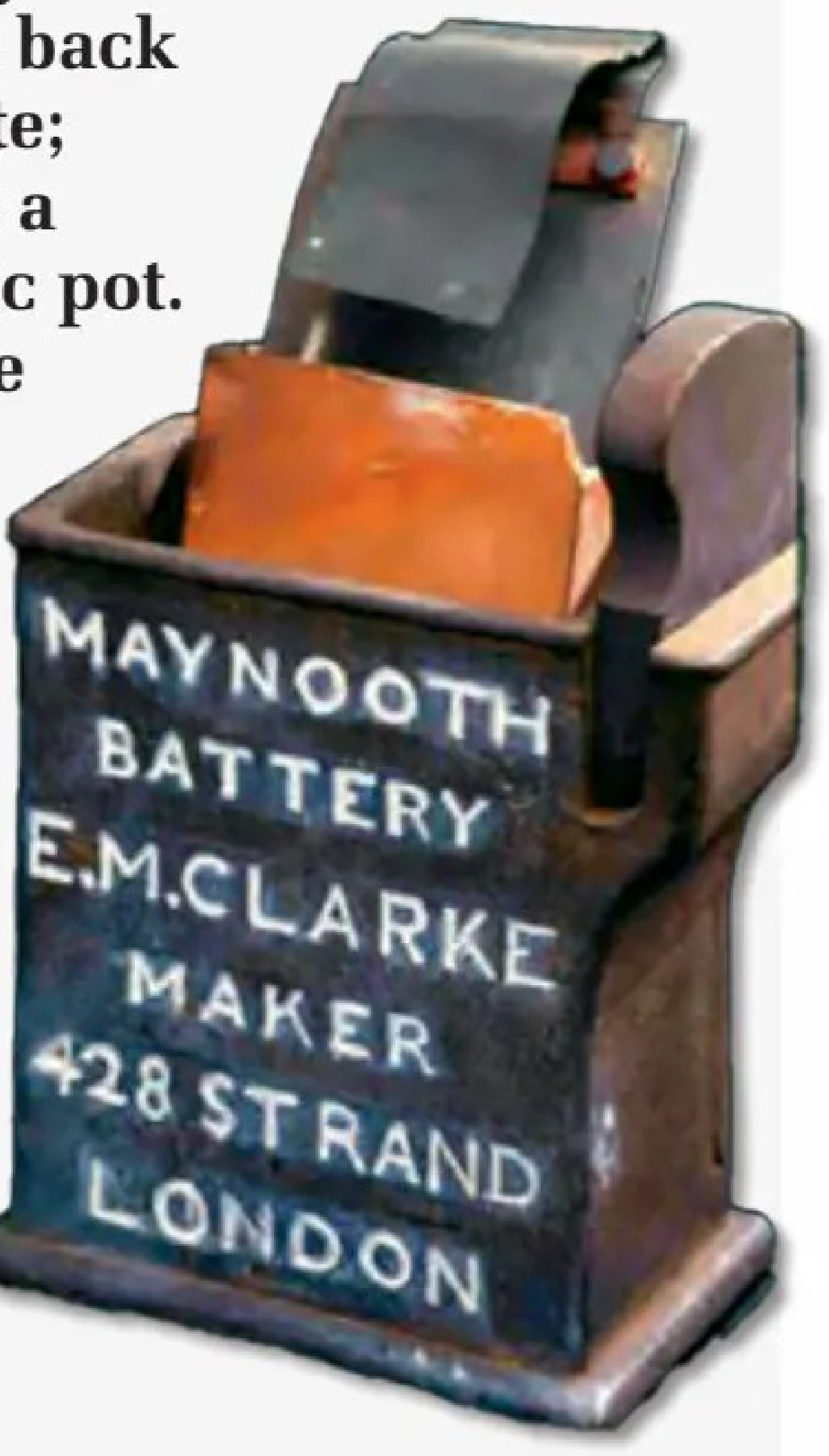


Fig.13: the Maynooth battery. At the back is the zinc plate; in front of it is a porous ceramic pot. Both are inside the iron container, which forms the other plate. *Source: Maynooth College Museum –* pemag.au/link/abp7

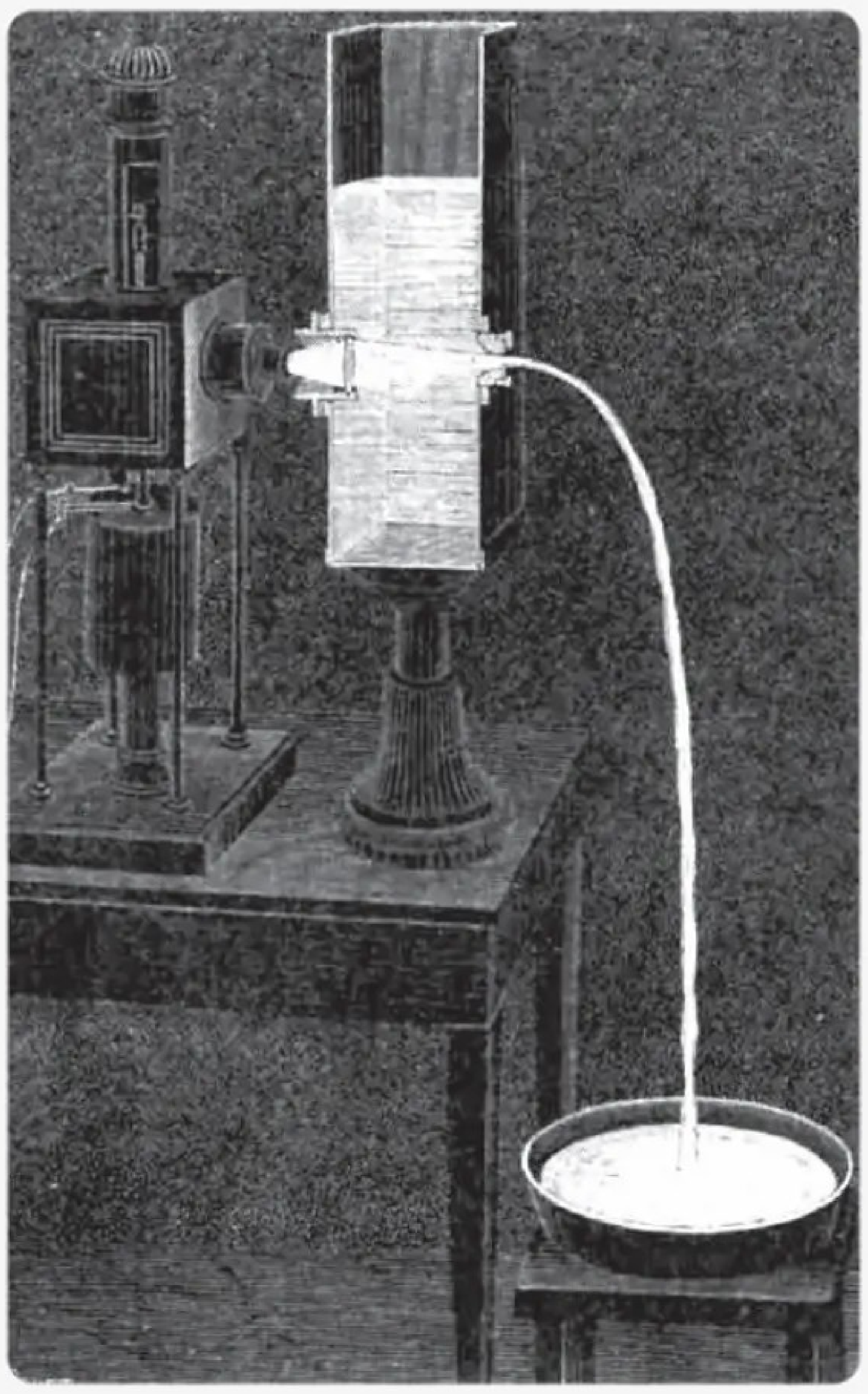


Fig.14: Jean-Daniel Colladon's experiment demonstrating total internal reflection in a stream of water. *Source: La Nature magazine, 1884.*

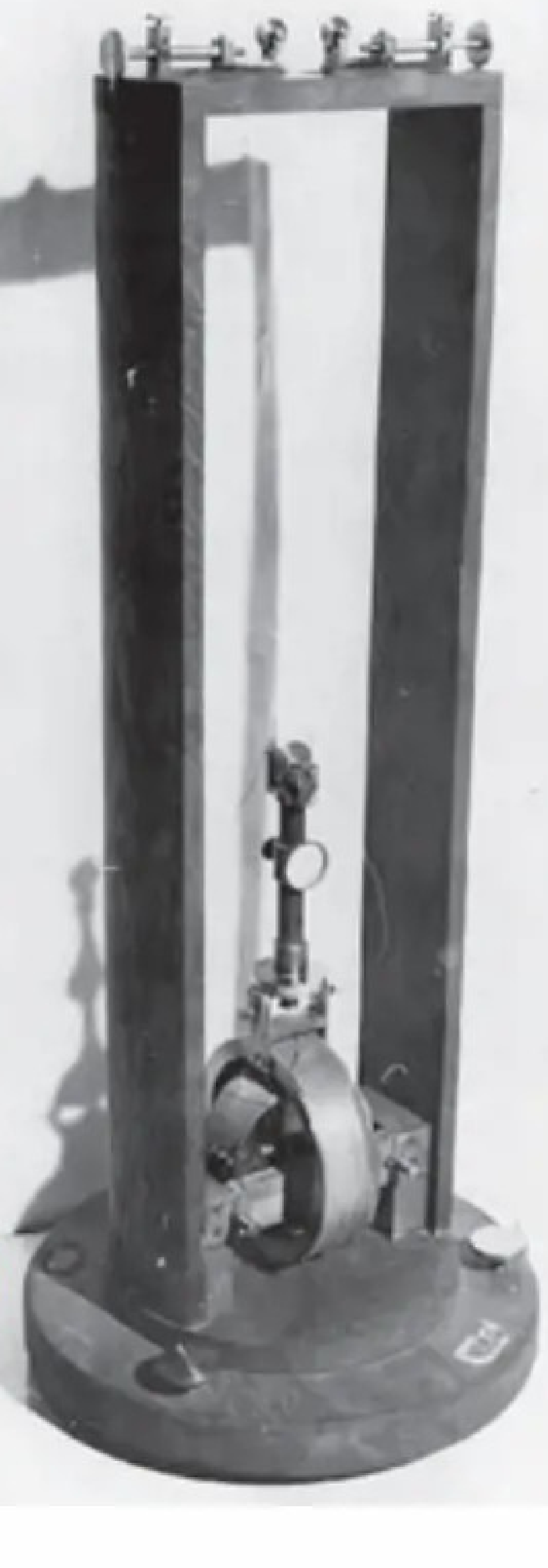


Fig.15: a replica of Weber's electrodynamometer made in 1961. *Source:* https://americanhistory.si.edu/collections/search/object/nmah_1273644

"Maynooth battery" (Fig.13) from iron and zinc, with 136L of acid and 577 individual cells. Back then, there was no way to measure voltage or current, so he measured the lifting capacity of an electromagnet to test its relative power.

James Bowman Lindsay 1799-1862
incandescent light globe

Invented the first incandescent light globe in 1835, enabling him to "read a book at the distance of 1½ foot", but he never patented it and did not receive credit. In 1845, he suggested that telegraphy could work across water, including the Atlantic. He proposed welding to join the cables and sacrificial anodes for corrosion protection.

Frederick Collier Bakewell 1800-1869
fax machine

Demonstrated an "image telegraph" machine in 1851, an early fax machine and an improvement upon the system of Alexander Bain. The system worked by drawing on metal foil using insulating ink. The foil was rolled into a cylinder, and a stylus read the conducting and insulating areas, converting them into signals to be transmitted.

The image was reconstructed on treated paper that electrical impulses could discolour. Keeping appropriate synchronisation at both ends was difficult, and the system was never commercialised.

Moritz Hermann 1801-1874
Jacobi's law

Also known as Boris Semyonovich (von) Jacobi, invented a process for making printing plates by electroplating in 1838. In 1839, he made an 8.5m-long battery-powered boat that carried 14 passengers.

He studied electric motors and, in 1840, published the maximum power theorem or **Jacobi's law**, which states that for maximum power transfer, the load resistance must match the source resistance. He also worked on the development of the electric telegraph during 1842-1845.

Charles Wheatstone 1802-1875
telegraph and Wheatstone bridge

He performed an experiment in 1834 to determine the "velocity of electricity". His result was about 50% too high. In 1837, Wheatstone also began work with William Fothergill Cooke on the telegraph. In 1843, he improved and popularised Samuel Hunter Christie's "diamond method", which became known as the Wheatstone Bridge.

Jean-Daniel Colladon 1802-1893
total internal reflection (TIR)

Demonstrated total internal reflection in a falling stream of water in 1842 (an experiment which can be done at home) – see Fig.14. This allowed optical fibres to be developed much later. The original idea was used to illuminate water fountains such as at the Paris World Exposition of 1889.

Frederick de Moleyns 1804-1854
platinum filament incandescent light globe

He obtained the first patent for an incandescent light globe in 1841. It used a platinum filament, although he also experimented with carbon filaments.

Emil Lenz 1804-1865
Lenz's law, resistive heating and electroplating

Formulated **Lenz's law** in 1834, which specifies the direction of a current induced by a magnetic field. He also independently discovered **Joule's law** (or the **Joules-Lenz law**) in 1842, which describes how an electric current causes a conductor to heat, otherwise known as resistive or ohmic heating.

He also participated in the development of electroplating with his friend Moritz Hermann.

Louis Breguet 1804-1883
Foy-Breguet telegraph

Developed a needle telegraph in 1842, the Foy-Breguet telegraph, used on the French railways and in Japan. In 1847, he suggested using finer diameter wires to protect telegraph wires against lightning strikes, the predecessor of the fuse.

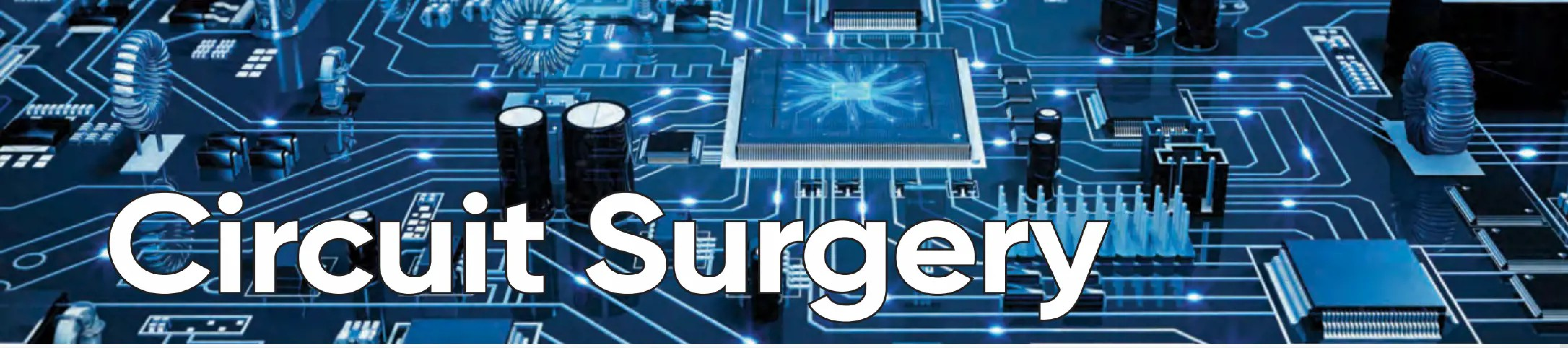
Wilhelm Eduard Weber 1804-1891
electrodynamometer

Together with Carl Gauss, he built the first working electric telegraph, nearly 1.6km long, in 1831. Weber developed many sensitive devices for detecting and measuring electric currents and magnetic fields, including precise measurements of the Earth's magnetic field.

He also invented the electrodynamometer (Fig.15), a device that can measure current, voltage or power via the interaction of magnetic fields through two coils. This device was used to validate Ampère's force law experimentally. The SI unit of magnetic flux, the **weber (Wb)**, is named after him. For more on Weber, visit: pemag.au/link/abn7

Next month

That's all we have room for this month. The second article in this series, to be published next month, will continue the chronological list of early inventors. PE



Circuit Surgery

Regular clinic by Ian Bell

Topics in digital signal processing – Signal filtering in the digital domain

We are looking at various topics related to digital signal processing (DSP).

DSP covers a wide range of electronics applications where signals are manipulated, analysed, generated, stored or displayed as digital data but originate from and/or are converted to real-world signals for interaction with humans or other parts of the physical world.

Fig.1 shows the key elements of a generic DSP system with a signal path from an analog input via digital processing to an analog output. This does not necessarily represent every DSP system (not all have all the parts shown), but it serves as a reference for the various subsystems we will consider.

Last month we discussed some basic concepts of digital signal processing and used a moving average function as an example. This month, we will continue to explore the moving average filter simulation and will also discuss the general structure of digital filters.

Frequency responses in LTspice

Last month, we obtained the frequency response of a five-point moving average digital filter using behavioural simulation in LTspice. LTspice may not be the most obvious tool to use for digital filters. However, with the help of its transient frequency response analysis (FRA), we obtained the response of a full system comprising sampling, anti-alias and reconstruction filtering and the moving average calculation.

The behaviour of the digital filter was implemented with the help of the delay function available from LTspice's behavioural voltage (and current) source. A delay applied to a (sampled) input signal is equivalent to holding sample data in memory in a digital circuit or computer.

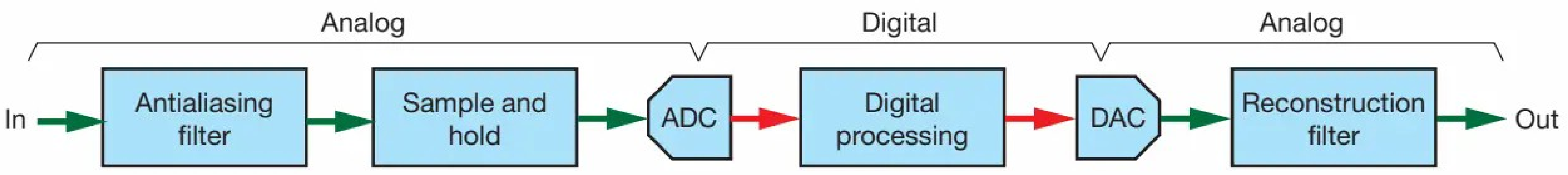
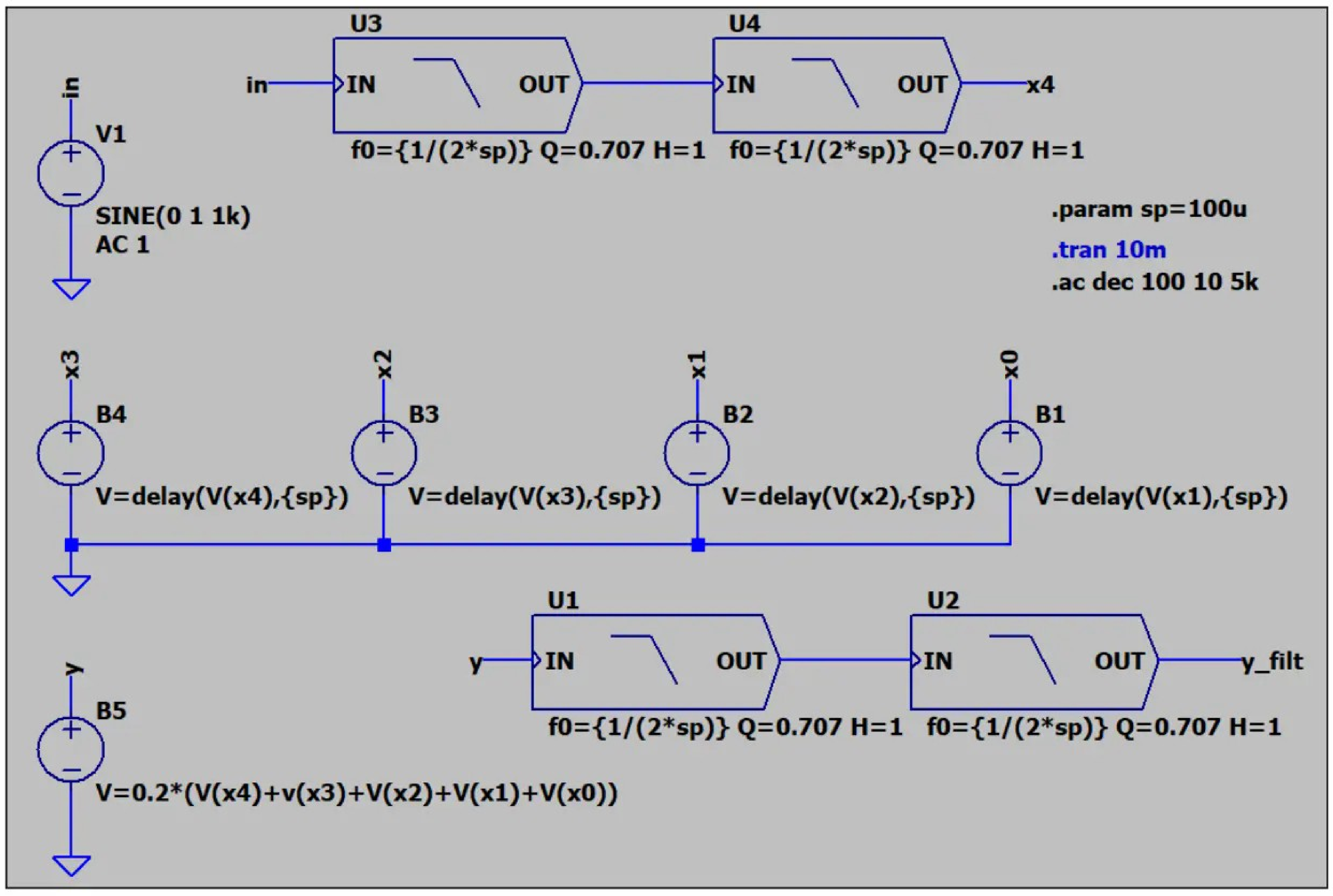


Fig.1: a generic digital signal processing (DSP) system structure.



```
.param sp=100u
.tran 10m
.ac dec 100 10 5k
```

Fig.2: a behavioural model in LTspice of a moving-average filter without sampling.

Typically, digital filters perform operations on multiple samples, which is equivalent to a set of delays equal to integer multiples of the sampling period. This means we can use the signals obtained by a chain of delays in LTspice as the equivalent to the set of samples held in a DSP system's memory.

We can then use another behavioural source to implement the maths performed by the filter (in this case, taking the average of the five most recent input samples).

The FRA in LTspice is a relatively new addition; it is useful because it allows us to obtain a frequency response in cases where AC analysis does not work, such as switched circuits. As we saw last month, AC analysis failed for the full moving average filter due to the sampling operation.

FRA is relatively difficult to set up as it requires components to be added to the schematic and configured. For the example last month, it required a fairly cumbersome manual addition of analysis frequencies to obtain an accurate response graph. The FRA was designed for stability analysis of switched mode power supplies, so it is not optimised for what we were doing.

AC analysis revisited

If we only want the frequency response of the filtering operations, not the full sampling system, we can use an AC analysis by removing the sampling part of the system. We can include the anti-aliasing and reconstruction filtering, or not, as required. AC analysis of just the digital processing model provides the response of the filter itself, without the impact of the sampling and other filters.

The LTspice schematic in Fig.2 is a version of the five-point moving average filter from last month without the sampling system. This allows us to run a standard AC analysis. As with the FRA, we cover the range of 100Hz to 5kHz, which is up to the Nyquist limit (half the sampling rate).

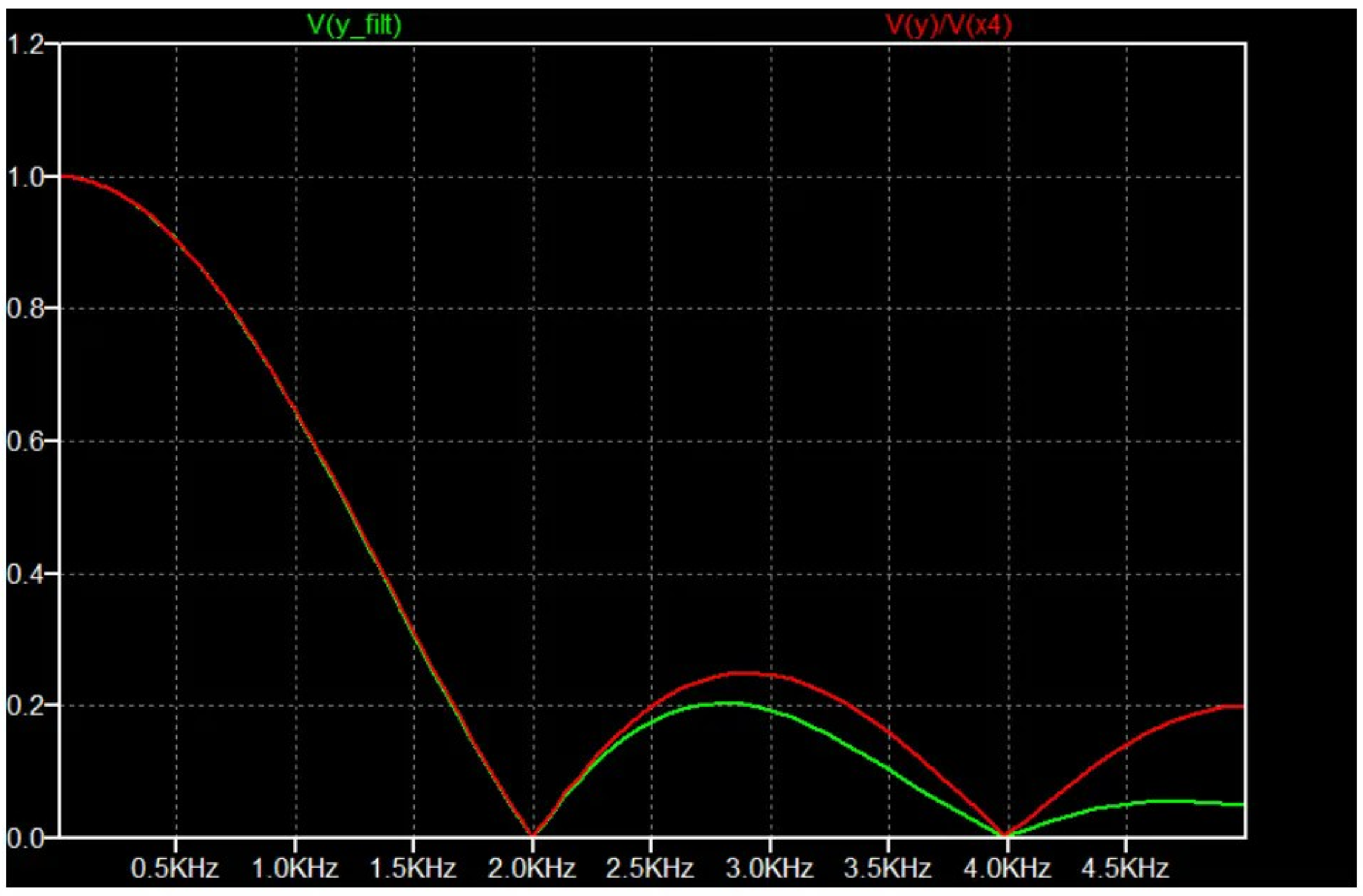


Fig.3: the simulation results for the Fig.2 schematic.

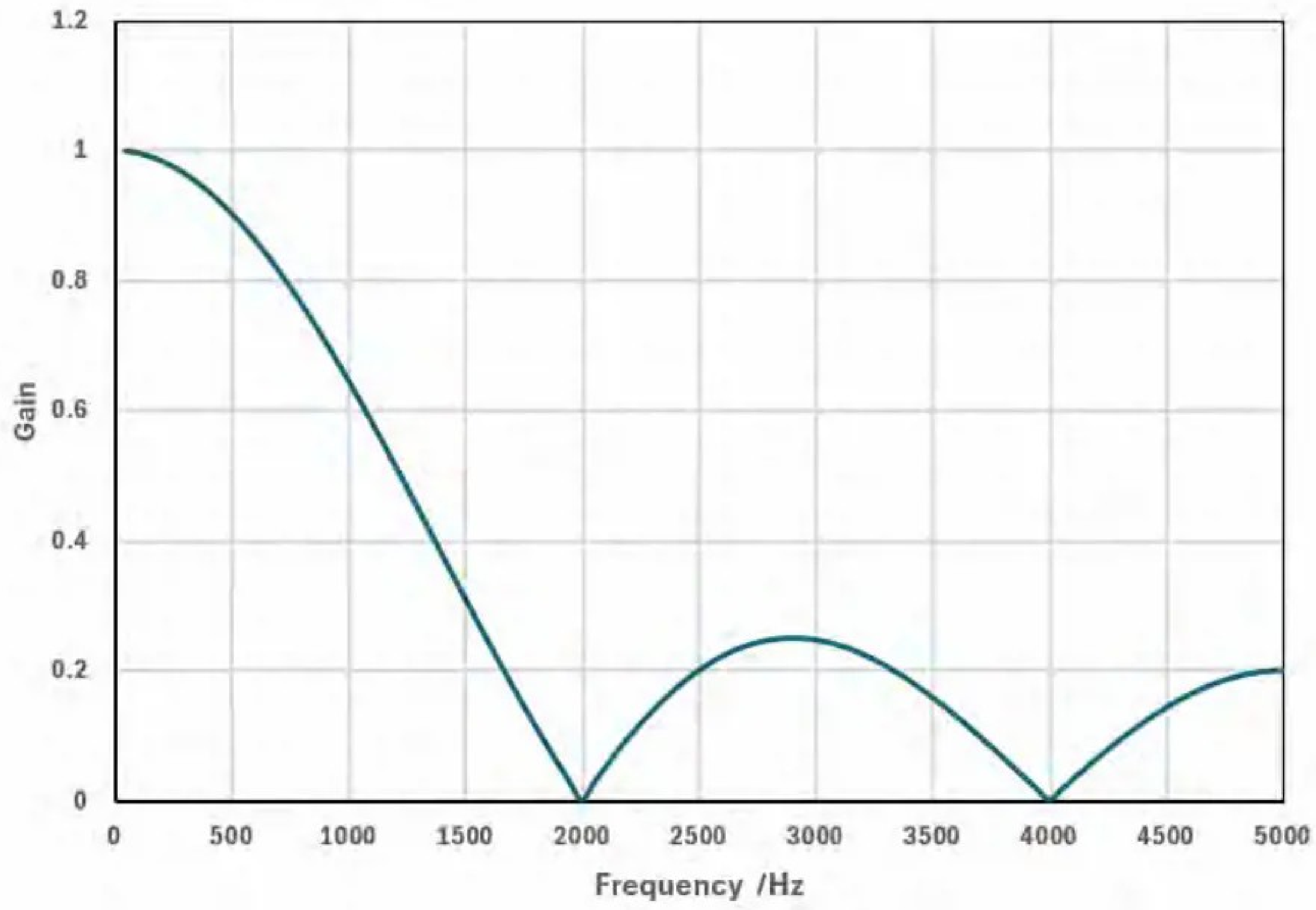


Fig.4: the calculated response of a 5-point moving-average filter.

The results are shown in Fig.3. The green trace shows the response of the system, including the antialiasing and reconstruction filters. The red trace shows the response of just the moving average filter by plotting V(y)/V(x4), that is, the gain from the input to the delay chain (on net x4) to the output of the source calculating the average (on net y).

Comparing the two traces shows the impact of the antialiasing and reconstruction filters on the overall response – this is a reduction in gain as frequencies get closer to Nyquist rate and the filters have more impact. The response from FRA analysis (shown last month) is different from the two traces in Fig.3 because it also includes the effect of processing a sampled signal.

The theoretical response of the moving average filter (as discussed last month) is shown in Fig.4. It closely matches the trace for the response of the moving average filter on its own from Fig.3 (red trace). This is as expected because this part of the LTspice schematic is directly performing the mathematics of the moving average function without any non-ideal circuit effects or additional influences.

The simulations discussed last month and above show that LTspice can be used to obtain the frequency response of a basic digital filter with or without the effects of other parts of the system.

We can also run transient analysis simulations to look at circuit waveforms. Last month, we pointed out that the moving average filter is not particularly good as a low-pass filter but is useful for removing noise from pulse-based signals. Therefore, we will use this as the basis of an example transient simulation.

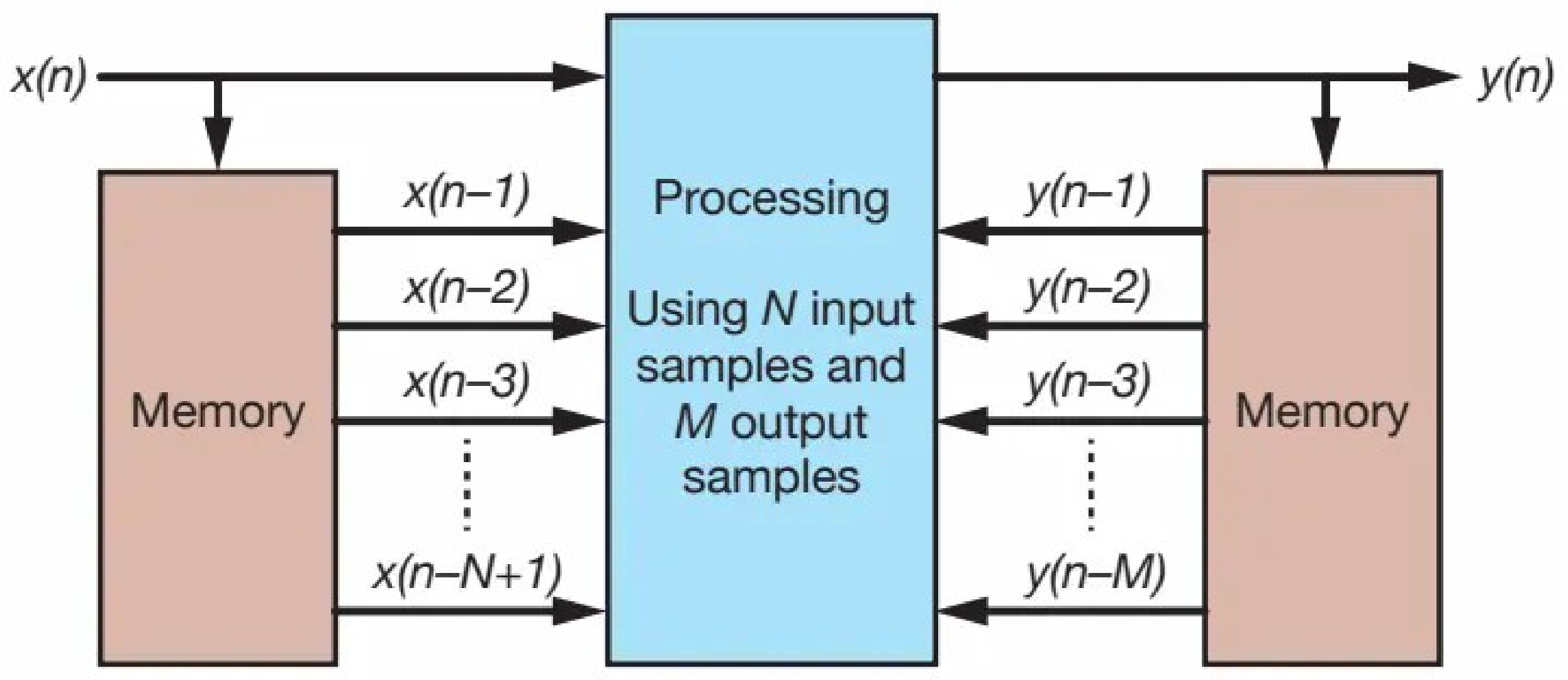


Fig.5: the general structure of a digital signal processing system.

The noise filtering example will work better with a filter averaging over more samples, which raises the issue of defining the function of the filter more effectively in LTspice – editing large expressions on the schematic is not particularly easy. This means we should define the general structure of digital filters in more detail, so we will discuss that before returning to the example.

Digital filter structure

The general structure of a digital signal processing system was discussed last month and illustrated in Fig.5. This is labelled a little differently from last month to show *N* and *M* as the number of input and output samples used by the processing, respectively. This is for compatibility with the discussion to follow.

To recap, the two memories shown in Fig.5 hold past input (*x*) and output (*y*) sample data, which can be used in the calculations that deliver the required function. Not all systems have output memory and therefore obtain the output as a function of just the current and past input samples. This includes the moving average filter we have been discussing.

Last month, we wrote the five-point moving average filter function as:

$$y(n)=\frac{x(n)+x(n-1)+x(n-2)+x(n-3)+x(n-4)}{5}$$

Recall that $x(n)$ is the current input, $x(n\text{-}1)$ the previous input and so on. The average is formed in the standard way: add up all the values and divide by the number of values (five in this case). We can also write the same function as:

$$y(n)=0.2x(n)+0.2x(n-1)+\\0.2x(n-2)+0.2x(n-3)+0.2x(n-4)$$

The filter output sample is calculated by multiplying each sample by 0.2 and adding up the resulting values. Written in this form, it is clear that in general we do not have to multiply all the samples by the same value. If we use different values, we no longer have a moving average filter; by choosing suitable values, we can create filters with different characteristics. We will return to discuss this aspect of digital filters next month.

Assuming we multiply each sample by a different value, we can write a general equation for a filter using five input samples as:

$$y(n)=a_0x(n)+a_1x(n-1)+\\a_2x(n-2)+a_3x(n-3)+a_4x(n-4)$$

The values a_0, a_1, … a_4 are referred to as the coefficients of the filter. The collection of coefficients is also referred to as the filter kernel.

We do not have to use exactly five samples, of course, so we can further generalise to using *N* samples (as depicted in Fig.5). In this case, each term in the summation is of the form $a_ix(n\text{-}i)$, where *i* is an integer ranging from 0 to *N*-1. We can then write:

$$y(n)=\sum_{i=0}^{N-1} a_i x(n-i)$$

With *N*=5, this is equivalent to the previous equation. The symbol Σ is the summation operator, which we met in the August 2024 *Circuit Surgery* article. It provides a compact way of writing the summation of the $a_ix(n\text{-}i)$ terms.

In the August issue, we discussed the mathematical operation called convolution. If you recall that or are already familiar with the mathematics, you will see that the above equation is a convolution of the coefficients with the samples.

The above equation allows us to draw a more detailed version of Fig.5 that depicts the actual processing being performed in a filter (Fig.6). The memory is shown as a chain of delays, each equal to the sampling period. These are labelled Δ*t*, with the Greek letter delta (Δ) for difference and *t* for time (a delay is a time difference).

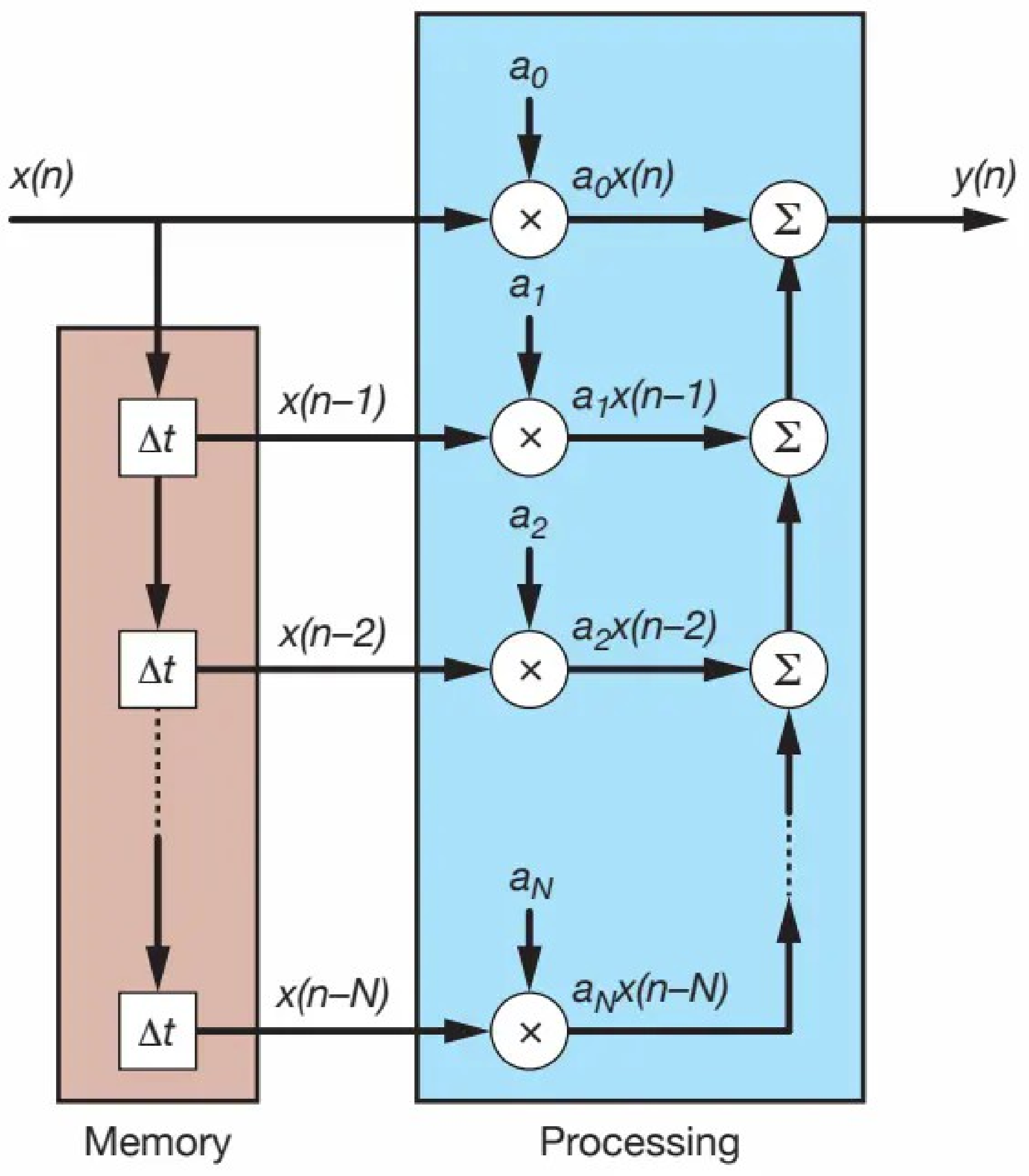


Fig.6: the structure of a finite impulse response (FIR) filter.

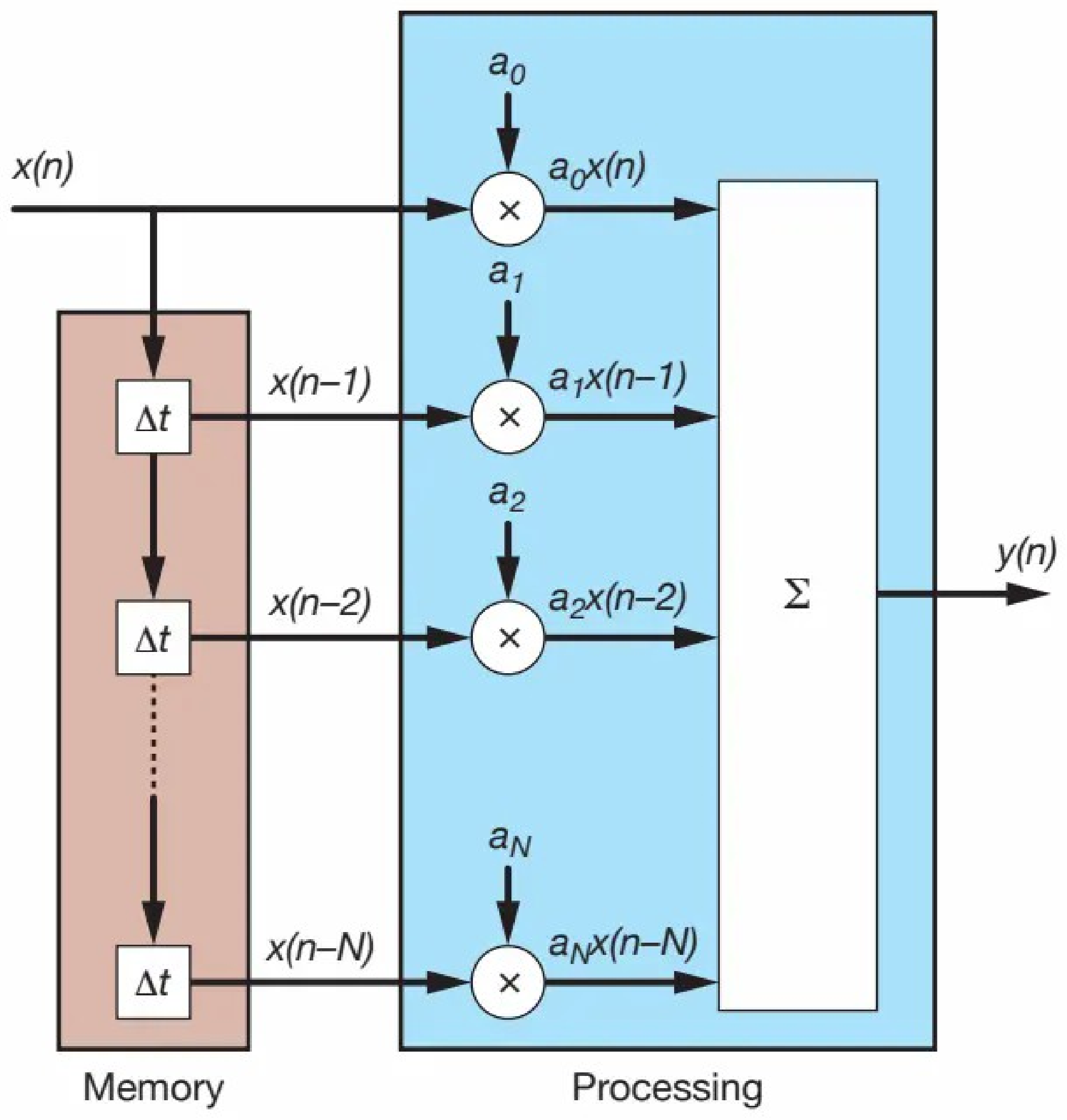


Fig.7: a finite impulse response filter with alternative depiction of the addition part.

The structure of the memory section is very similar to the delay chain (B1 to B4) in the LTspice model in Fig.2. The Δt delay blocks can be implemented as registers in a (multi-bit wide) shift register in a digital circuit clocked at the sample rate, or as locations in a buffer memory used to store the samples in a software implementation.

If you look at similar diagrams to Fig.6 on websites or in textbooks, the delay blocks will often be labelled with z^{-1} rather than Δt. This relates to the use of a mathematical technique called the Z-transform. This is similar to the (probably better known) Laplace transform.

In both cases, time domain representations of signals and circuit responses are converted to an alternative form (called the Laplace domain and z-domain), facilitating analysis and design. In both cases, the alternative domain is based on a complex-number representation of frequency (not simple frequencies in Hz).

The Laplace domain is used for continuous time signals and the z-domain for sampled signals. In the z-domain, multiplying by z^{-n} creates a delay of n sampling periods, hence the labelling of a single sample delay block with z^{-1}.

We will not investigate the Z-transform in detail due to the strong maths involved, but if you are new to DSP it useful to know it exists and may merit further investigation. For commonly used filters, you do not have to develop theory from scratch using these transforms to produce designs, and there are tools available to assist with calculations.

In Fig.6, the circular blocks labelled × are multipliers that multiply the samples by the relevant coefficients. This could be performed by hardware multipliers or in software. The circular blocks labelled Σ are adders that perform the summation of all the $a_i x(n\text{-}i)$ terms. Again, these could be hardware adders or software operations.

The structure in Fig.6 is of a chain of adders, which corresponds with a sequence of operations in software or circuit structured that way. An adder circuit could be structured differently (eg, as a tree of adders rather than a chain). The diagram in Fig.7, which shows all the terms being added but with no specific adder arrangement, is also commonly used to show the structure of this type of filter.

The general system structure shown in Fig.5 includes the processing of output samples. Digital filters can also use output samples similarly to the use of input samples already discussed. That is, output samples can be multiplied by a coefficient (b_n) and included in the summation that produces the output.

The structure of such a filter is shown in Fig.8, in which M previous output samples are used in the calculation of the filter's current output.

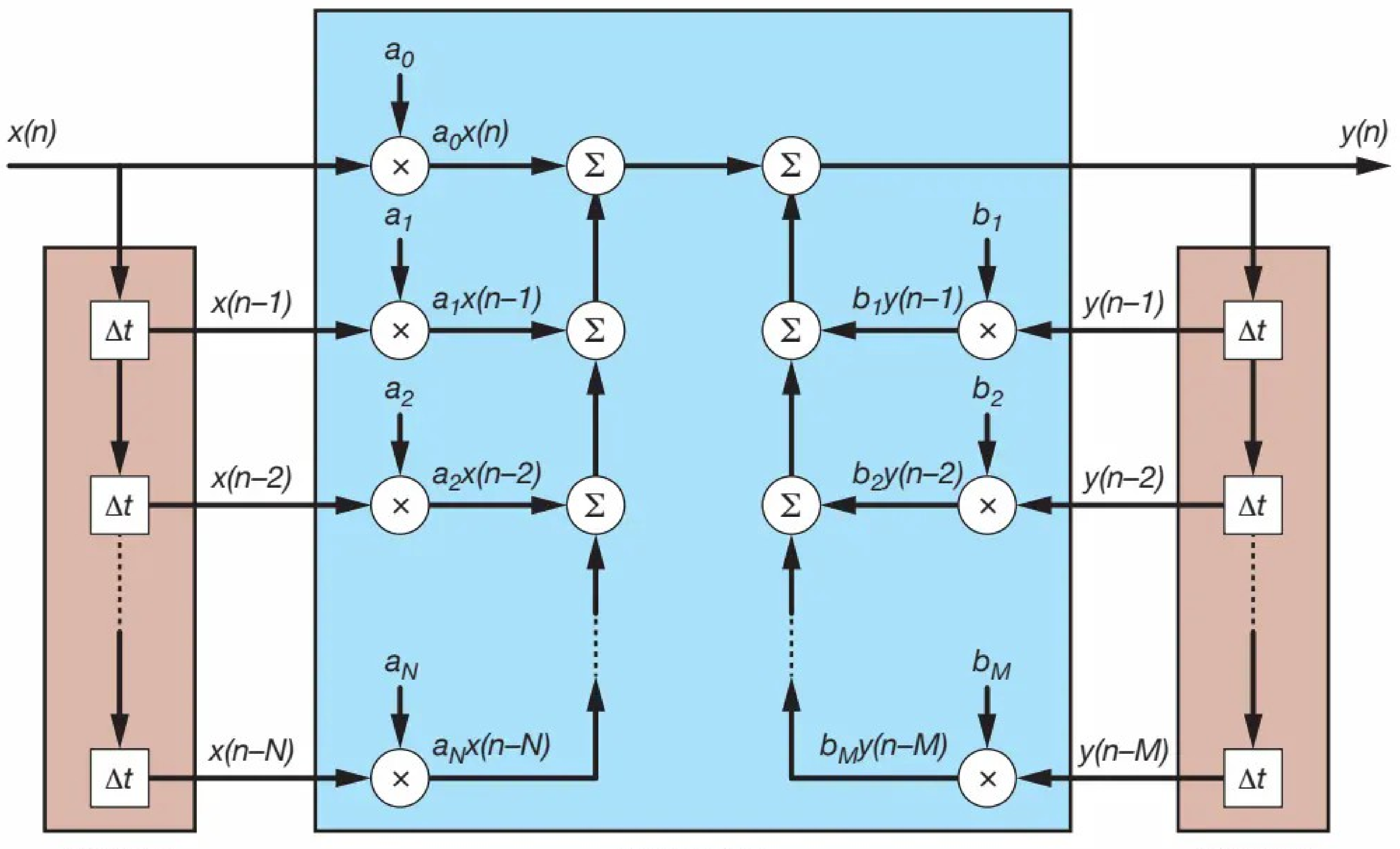


Fig.8: the structure of an infinite impulse response (IIR) filter.

Impulse response

We have discussed impulses in previous articles. For continuous time systems, an impulse is an infinitely short pulse of unit area, but for sampled signal processing, it is a single sample of value 1. It is useful to consider what happens if we apply an impulse to a filter – the result is called the impulse response, and signal processing theory shows that this defines the behaviour of the filter.

Consider an impulse input to the filter in Figs.6 & 7. Initially, all the samples in memory and the input $x(n)$ will be zero. For one sampling period, $x(n)$ will change to 1 and then return to 0 for all time. When $x(n)=1$, the output $y(n)$ will be $a_0x(n) = a_0$, as all other samples in the memory at the time will be zero and will not contribute anything to the summation total.

On the next sampling period, $x(n)=0$ and the 1 will move to $x(n\text{-}1)$ in the memory structure. It will remain the only non-zero sample value contribution to the summation. So, the output will then be $a_1x(n) = a_1$.

On the next sampling period, the 1 will move to the $x(n\text{-}2)$ memory location and, following the same pattern, the output will be equal to a_2. Thus, with an impulse input, the filter will deliver the values of its coefficients in sequence, so the impulse response clearly defines the filter (the coefficients are key to the specific response).

After all the coefficients have been fed out, the output will then be zero for all time. Thus, the period over which the filter produces a non-zero output from an impulse input is finite. This type of filter is therefore called a Finite Impulse Response (FIR) filter.

The filter structure shown in Fig.8 has a more complex behaviour. Past output samples are fed back into the filter, so once the output has taken a non-zero value, it will circulate around the output sample processing loop for all time. Therefore, filters with this structure will produce an infinitely long non-zero output from an impulse input and are referred to as Infinite Impulse Response (IIR) filters.

Real IIR filter implementations may return to zero output after an impulse input due to numerical precision limitations.

LTspice functions

Having discussed digital filter structures in general terms, we will return to our LTspice simulations. The previous example had five coefficients, but real filters may have many more, and all the coefficients all may be different. Also, for accurate operation, the coefficients need to be defined to a good number of decimal places.

This requires a lot of text to define the filter function (eg, the expression for **V** in the B5 source in Fig.2), which is difficult to edit in the source value dialog.

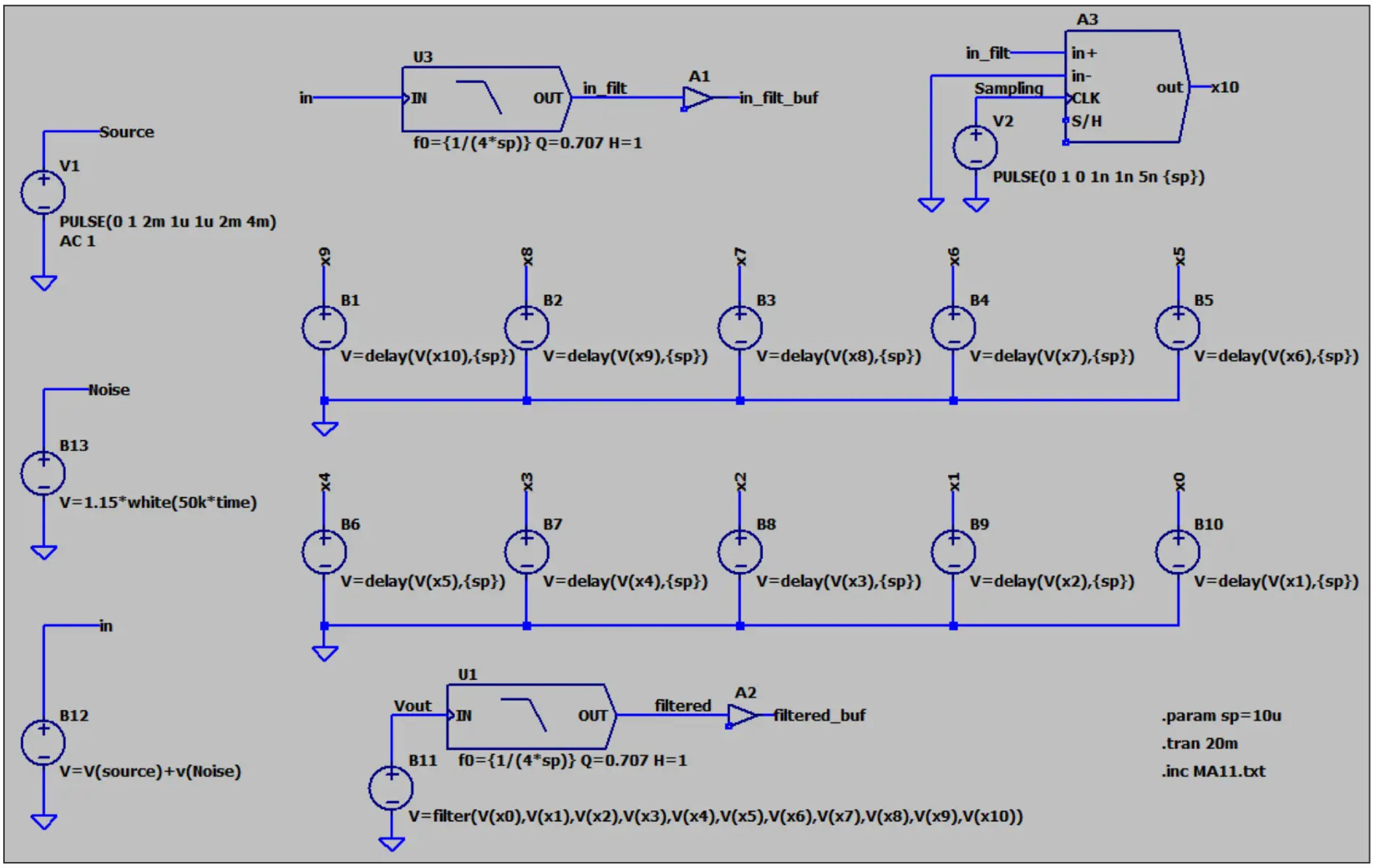

U3
in
IN
OUT
in_filt
f0={1/(4*sp)} Q=0.707 H=1
A1
in_filt_buf
A3
in_filt
in+
in-
Sampling
CLK
S/H
out
x10
V2
PULSE(0 1 0 1n 1n 5n {sp})
Source
V1
PULSE(0 1 2m 1u 1u 2m 4m)
AC 1
x9
B1
V=delay(V(x10),{sp})
x8
B2
V=delay(V(x9),{sp})
x7
B3
V=delay(V(x8),{sp})
x6
B4
V=delay(V(x7),{sp})
x5
B5
V=delay(V(x6),{sp})
Noise
B13
V=1.15*white(50k*time)
x4
B6
V=delay(V(x5),{sp})
x3
B7
V=delay(V(x4),{sp})
x2
B8
V=delay(V(x3),{sp})
x1
B9
V=delay(V(x2),{sp})
x0
B10
V=delay(V(x1),{sp})
in
U1
Vout
IN
OUT
filtered
A2
filtered_buf
B12
V=V(source)+v(Noise)
B11
f0={1/(4*sp)} Q=0.707 H=1
V=filter(V(x0),V(x1),V(x2),V(x3),V(x4),V(x5),V(x6),V(x7),V(x8),V(x9),V(x10))
.param sp=10u
.tran 20m
.inc MA11.txt

Fig.9: an LTspice simulation of a moving-average filter applied to a noisy pulse waveform

Behavioural sources can be defined in terms of numerous built-in functions, but it is also possible to write your own functions using the `.func` directive. We can use this feature with the built-in `table` function to define a look-up table of coefficients. For example, we can define a function called `a` that defines five coefficients of values 0.2, 0.4, 0.5, 0.3 and 0.1 as follows:

```
.func a(i){table(i,0,0.2,1,
+  0.4,2,0.5,3,0.3,4,0.3)}
```

As with any SPICE directive, this can be placed on the schematic. The `table` function uses the first parameter (i in this case, which is passed to the function) to look up values defined by the number pairs given in the remaining parameters. So, for example, using `a(2)` in a behavioural source expression would be equivalent to 0.5, and `a(0)` would be 0.2.

The table function can actually interpolate between the defined values, but this is not needed for coefficient look-up.

Writing the function with one coefficient on each line makes reading and editing it easier, so a better format is:

```
 .func co(i) {table(i,
 + 0,0.2,
 + 1,0.2,
 + 2,0.2,
 + 3,0.2,
 + 4,0.2)}
```

This is the same code with extra line breaks. We must put a + character at the start of each new line of the function; this is called a line continuation.

SPICE directives must normally be on a single line, but line continuation provides a means of formatting a single line across multiple lines of text as viewed by the human reader. The compiler treats lines starting with + as part of the previous line.

With the coefficient function defined, we can write another function to implement the filter, for example, with five coefficients and five samples (`x0` to `x4`) passed to the function:

```
.func filter(x0,x1,x2,x3,x4) {
+ a(0)*x4+a(1)*x3+a(2)*x2+
+ a(3)*x1+a(4)*x0 }
```

Again, this can be put in the schematic, but for larger numbers of coefficients, it becomes easier to use a separate text file instead. To do this, put the code for both functions in a single text file and use the SPICE `.inc` directive to use it as part of the simulation. For example, if the file name is "filter_def.txt", put the SPICE directive `.inc filter_def.txt` in the schematic (as a directive, not just text).

The behavioural source implementing the filter output is then defined using the filter function, for example:

```
V=filter(V(x0),V(x1),V(x2),
+ V(x3),V(x4))
```

Here, `V(x0)` to `V(x4)` are the input sample voltages after the various delays, as in Fig.2.

Transient filter simulation

The schematic shown in Fig.9 is similar to Fig.8 from last month. It is a moving-average filter using 11 rather than five samples.

The coefficients and filter function are defined in the file "MA11.txt", which follows the pattern of the code discussed above. The 11 coefficients are all equal to 0.0909091 (an approximation of 1/11).

The simulation includes the sampling process (unlike Fig.2, although we did include it in last month's circuit). The sampling period is 10µs (equivalent to a 100kHz sampling frequency). The reconstruction and antialiasing filter cutoffs are set to $f_s/4$, which is 25kHz, well below the Nyquist frequency limit (50kHz) to allow for the filter's roll-off.

The input signal is a pulse waveform with noise added to it. A clean pulse waveform, a 1V square wave with 4ms cycle time (2ms pulses) is created by source V1, and behavioural source B12 adds random noise to this using the **white** function.

The **white** function produces random number between -0.5 and 0.5, smoothly transitioning between values. Here, it is configured to change the voltage every 1/50k seconds (20us). This is achieved

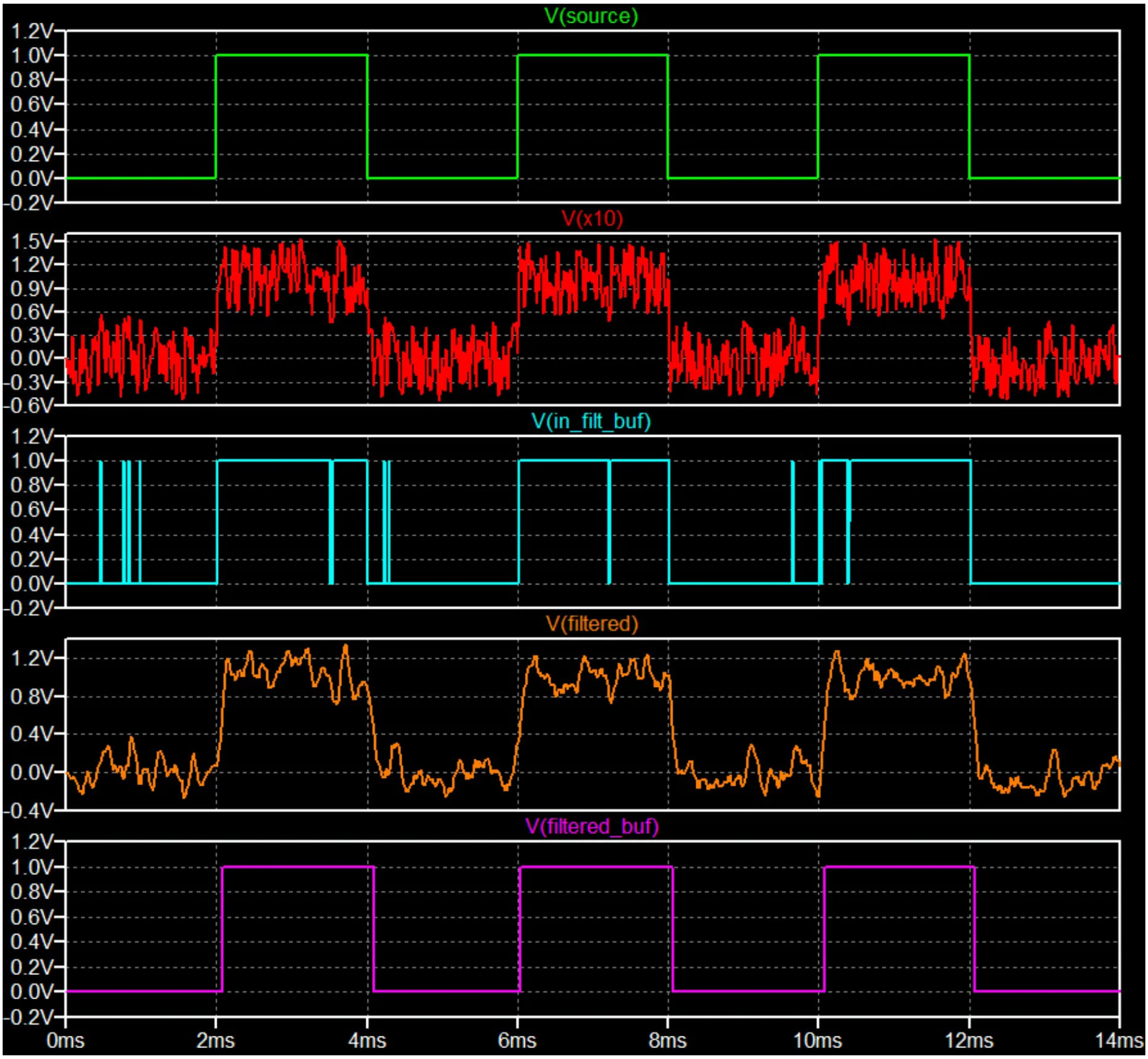


Fig.10: the results of the LTspice simulation of the configuration shown in Fig.9.

by passing **`50k*time`** to the **`white()`** function.

The maximum noise amplitude is larger than the pulse amplitude, which produces a signal that would likely be difficult to use without filtering. Specifically, the largest peak-to-peak output possible from the noise source is 1.15 V (due to **`1.15*white()`** in the expression for **`V`**), but most of the time the amplitude is lower.

The harmonics of the square wave are very small beyond the Nyquist limit, being 1/200th of the signal amplitude or lower. The noise voltage changes only every 20µs, which means that the noise spectral amplitude is also reducing above the Nyquist frequency. These factors, combined with the filter, prevent too much aliasing.

Components A1 and A2 are logic buffer gates with logic thresholds of 0.5V and 1.0V (the default for LTspice logic gates). These are connected to the input (A1) and filtered output (A2) respectively of the moving average filter.

The setup in this example is not designed to meet any specific use-case or design requirement. It is just configured to produce some example waveforms and illustrate how the sampled signal processing function can be included in an LTspice transient simulation.

The results are shown in Fig.10. The upper trace (green) is the clean input square wave. The second trace (red) is the sampled square wave plus noise at the input to the moving average filter (signal **`x10`**). The fact that this is sampled cannot easily be seen at this zoom level.

The third trace (cyan) shows the output of the logic buffer with **`x10`** as the input. Ideally, this should be equal to the source signal, but the noise causes unwanted switching (glitches). The fourth trace (orange) is the reconstructed output of the moving average filter, illustrating the reduction in noise with respect to the input.

The bottom trace (magenta) is the output of the logic buffer connected to the filter output. This is very close to the source signal (no glitches), with some delay due to the processing time of the filter. PE

Simulation files

Most months, LTSpice is used to support descriptions and analysis in Circuit Surgery.
The examples and files are available for download from the PE website:
https://bit.ly/pe-downloads

Techno Talk

Max the Magnificent

Testing Times

When I was young, bright-eyed, and bushy-tailed, any test equipment worth having was too expensive for me to have. Today, by comparison, I'm amazed by the power, sophistication and affordability of the offerings available to hobbyists.

The topic of testing and the equipment we use to do so has been much on my mind of late. There are a couple of pieces of test gear I personally could not do without. The first is my trusty multimeter. Actually, "multimeters" would be more apt, because I have them all over the place.

There's one next to me on the desk here in my office, another on my desk in my study at home, one in my backpack for when I'm on the road and others lurking hither and thither. The amazing thing is how cheap they are these days (I'm talking about multimeters for hobbyists rather than professional-grade devices). You can pick up a very serviceable unit for less than £10.

Scope it out

Another piece of test equipment that provides myriad test functions is an oscilloscope. Back in the 1950s and 1960s, these devices were monstrous in terms of size, weight and cost.

In addition to a cathode ray tube (CRT) for display, their rugged metal cabinets contained a cornucopia of valve-based electronics. Those valves (also known as vacuum tubes) used to perform tasks like controlling the vertical and horizontal deflection of the electron beam(s), amplifying signals and implementing switching and timing circuits.

Oscilloscopes like that were still found in the engineering department when I started university in 1975. They were so heavy that you could give yourself a hernia lifting one, so we wheeled them around on trolleys.

Things didn't get much better in the 1980s, even though the valves were swapped out for transistors and silicon chips. Although these components were smaller and lighter, the designers of the scopes employed lots more of them to squeeze in more features and functions. It was like a 'functionality arms race' between the various vendors. Also, the CRT remained in the heart of the beast.

A good professional-grade four-channel oscilloscope could easily cost £10,000 back in the day, so this wasn't something you would expect to find on a hobbyist's desk. How things have changed. I have a four-channel Rigol DS1054 rated at 50Mz and 1GSa/s with a large 7-inch LCD screen sitting next to me on my office desk.

It's small at 13 × 7 × 5 inches (33 × 18 × 13 cm), light (6.6 pounds or 3kg), and affordably priced (currently £300 on amazon.co.uk). It also lets me take screenshots directly to a USB memory stick plugged into a port on the front; a feature worth its weight in gold.

Take a guess

I was once offered a position based on my stating that my modus operandi for debugging things was to start with a WAG ("wild-ass guess").

"Perhaps you would care to elaborate a little", said my interviewer. That was his big mistake. Much like my dear old mother, the real trick is getting us to **stop** talking.

I explained that I would start with the obvious options. If the system in question was totally dead, for example, I would first check to see that it was plugged in (powered up). However, if the system was up and running but functioning erratically or incorrectly, this was where I would take a WAG based on how the system was behaving coupled with my past experiences.

Even if I hadn't seen anything like the current behaviour before, there may be clues tickling my unconscious mind.

My justification was that, if it proved to be correct, depending on the size, complexity, and cost of the system, taking a WAG could save a lot of time, resources, frustration, and money. Contrariwise, if my WAG proved to be incorrect, little effort would have been expended and little time would have been lost. I could then proceed to perform a regular in-depth analysis and diagnosis.

I got the job (perhaps in a desperate attempt to make me stop talking)!

Experience doesn't hurt

If you look at my *Weird and Wonderful Arduino Projects* column, which starts on page 7, you'll see that a member of the PE community called Ian emailed me with an interesting problem. In his email, Ian said, "See the attached photo. I have checked this a million times and am convinced there are no errors".

I must admit that my first thought on reading this was, "Oh Ian, why dost thou tempt the fates so?" I quickly set up a Zoom call with my friend Joe Farr because this sort of thing goes much quicker when there are two of you doing it.

Between us, Joe and I have a lot of experience in debugging things. This is largely due to the mistakes we made when putting them together in the first place. This explains why the first thing we did was to check the power and ground connections to the integrated circuits (ICs) mounted on the breadboards in Ian's photo.

At least, we checked the ground connections. We would have checked the power connections… but there weren't any. I emailed Ian. He added the requisite wires and the circuit functioned as expected. Problem solved.

Why did Joe and I check the power and ground connections before all else? We will never tell!

Trust? I think not!

I'm working on a new project that uses a 9V DC external power source, which I need to drop to 5V inside the device. I can get by with 5V at 1A, but I'd prefer to have 3A or more to play with.

I could use a regular, cheap-and-cheerful linear regulator, but they tend to run hot as the power rises.

I found some small and tasty buck (step-down) regulator modules on Amazon here in the USA. With their 9-36V input and 5V @ 5A output, these sounded too good to be true. They were indeed (too good to be true).

Joe purchased some equivalents on Amazon in the UK (https://pemag.au/link/ac2m). His arrived before mine. Joe tested them using a bench power supply and an electronic load. Oh dear. The output voltage started to droop at 1A, it fell to 4.4V by 2A, and it collapsed to 0V shortly thereafter.

Thank goodness for test equipment, is all I can say!

Editor's note: even some switch-mode IC manufacturers rate their parts based on the peak current they can deliver, which can be considerably higher than the average! PE

Using Electronic Modules with Jim Rowe

Quason VL6180X

- Laser rangefinder
- Light level sensor

VL6180X
2v8
GPIO
SCL
VIN
GND
SHDN
SDA

This module should be of particular interest if you want to build robotic devices. It uses infrared (IR) light to accurately sense the proximity of objects from 0mm to well over 100mm. It's based on a technology known as FlightSense, patented by ST Microelectronics.

The Quason VL6180X range-sensing module comes on a tiny 17.8 × 20.3mm PCB with a handful of SMD components on it. As you can see from the photos, it includes three SOT-23-3 devices and one 12-lead SMD IC, itself only 4.8 × 2.8 × 1mm.

The secret is all inside that innocent-looking 12-lead IC in the centre of the PCB. There's a lot more in that tiny package than you might expect. It's a complete optical ranging system with a tiny IR (infrared) laser, two optical sensors (one for IR, the other for ambient light sensing), plus a microcontroller unit (MCU) with internal memory.

This IC is the heart of the VL6180X sensing module – the rest of the components are there just to support it.

Inside the VL6180X

It's made by European semiconductor manufacturer ST Microelectronics and uses its patented FlightSense technology.

Unlike optical sensors that attempt to detect distance by measuring the proportion of light sent to an object that is reflected back from it, ST's technology accurately measures the time the light takes to travel to the nearest object and reflect back to its sensor, which ST calls the 'time of flight'. In short, it's a kind of light-based radar or 'LIDAR'.

Fig.1 shows what's inside the VL6180X and should help in understanding how it works. Near the bottom is the MCU with its ROM (read-only memory) and RAM (random-access memory) below it, while above it is the ambient light sensing section. The IR laser driver section is shown in the centre, with the range detection section just above the MCU.

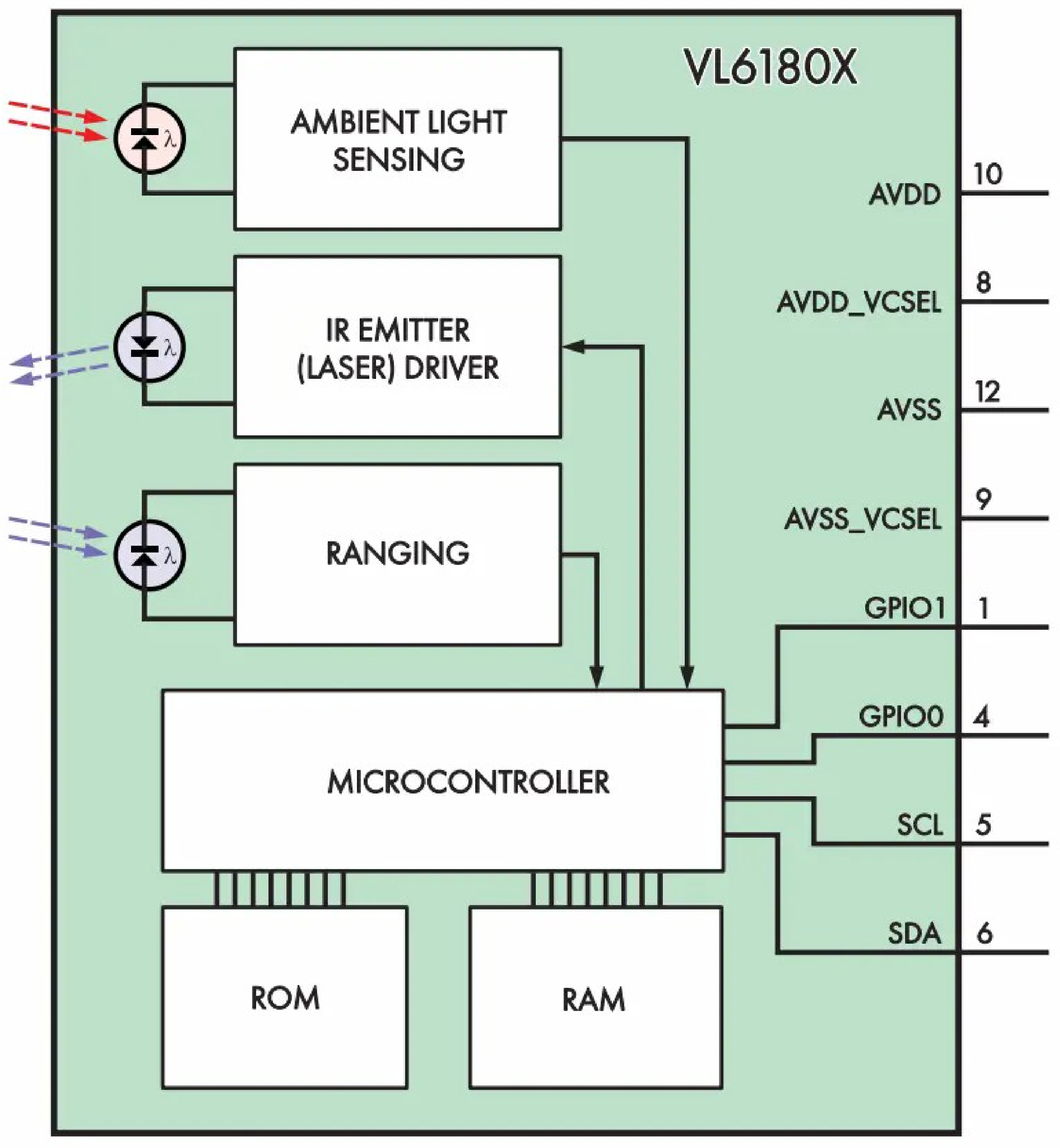
To make a ranging measurement, the MCU first sends a command pulse to the IR laser driver to send out a short IR light pulse at a wavelength of 850nm. Then, it measures the time until the ranging detection section reports that a reflected IR pulse has been received.

The MCU can then calculate the current distance to the object that reflected the IR pulse, by taking into account the speed of light in air and the time taken for the out-and-back journey.

The speed of light in air is close to 299,702,458m/s (metres per second), which equates to 299.702m per microsecond or 0.2997m per nanosecond. So light takes close to 3.336ns to travel one metre or 0.3336ns to travel 100mm. If the out-and-back journey of the light takes, say, 0.6672ns, the total path length is 200mm, so the distance between the sensor and the object must be 100mm.

The key to this method of determining distance is precise measurements of very short time delays. To measure over a range of 1-100mm with 1mm resolution, the chip must have a timer capable of measuring the difference between emission and reception from just 7ps (picoseconds) to 667ps with 7ps resolution or better. One picosecond is one trillionth (10^{-12}) of a second!

These capabilities are thanks to modern semiconductor manufacturing



NOTE: PINS 2, 3, 7 AND 11 HAVE NO INTERNAL CONNECTION

Fig.1: the block diagram for the VL6180X rangefinder IC. The internals appear quite simple, with a separate section for the light sensing, IR emitter and ranging. However, very precise timing is required to make calculations down to the millimetre resolution, so the actual circuitry is more complicated than you might think.

VL6180X
2v8
GPIO
SCL
VIN
GND
SHDN
SDA

Here the module is shown at nearly three times actual size for clarity.

techniques that can make tiny transistors with predictable properties.

In addition to this ‘time of flight’ range measurement, the VL6180X can also measure the ambient light level using the sensor and ambient light sensing (ALS) section shown at the top of Fig.1. This appears to be a ‘bonus’ feature as it does not factor into the distance measurements

It can measure light levels between 0.002 lux and 20,971 lux, with what is described as a ‘photopic’ response. That means it responds to light wavelengths in the visible range of 400-700nm (with a peak at around 550nm) as seen by the human eye at ‘well-lit’ lighting levels.

The MCU in the VL6180X can take these measurements either once or repetitively and can also interleave range and ALS measurements. It accepts commands and makes the measurement data available via the I²C port (pins 5 and 6) at lower right in Fig.1.

You are probably wondering how all these impressive things can be done by the very small and innocent-looking chip visible in the centre of the module PCB. Although they are not easy to see with the naked eye, there are actually three apertures on the top of the device, located on its centre line as shown in Fig.2 (which shows the top of the VL6180X at six times its actual size).

The largest aperture (0.58mm diameter) near the centre is for the ALS sensor, while the smaller 0.5mm diameter one near the far end is for the IR ranging laser emissions. The even smaller 0.3mm diameter aperture near the ALS at the pin 1 end is for the IR ranging return sensor. A fourth and very tiny ‘vent’ hole is at lower centre, midway between pins 3 and 4.

The VL6180X is designed to operate from a supply of 2.8V ±0.2V, with an average operating current of 1.7mA in ranging mode or 300μA in ALS mode. The current it draws in standby mode is less than 1μA. And the I²C interface can operate at up to 400kHz, with a 7-bit address of 0x29 (41 decimal).

The full module

Fig.3 shows the complete circuit of the Quason module, with the all-important VL6180X device (IC1) visible at lower left. At top centre is REG1, an XC6206 LDO voltage regulator used to step down the 5V input supply (at pin 7 of CON1) to the 2.8V needed by IC1. The 2.8V from REG1 is also made available at pin 6 of CON1, for possible use by external circuitry.

Useful links
- www.aliexpress.com
- www.st.com/content/st_com/en.html
- www.arduinolibraries.info/libraries/vl6180-x
- github.com/adafruit/Adafruit_VL6180X

Both the GPIO0 and GPIO1 pins of IC1 are pulled up to 2.8V via 47kΩ resistors. The GPIO1 pin is then taken directly to pin 4 of CON1, while the GPIO0 pin is connected to pin 3 of CON1 via diode D1. This allows IC1 to be held in standby mode by pulling pin 3 of CON1 to ground. That is why this pin of CON1 is labelled “SHDN” (for “shutdown”).

Mosfets Q1 and Q2, connected between the SCL and SDA pins of IC1 and the corresponding pins 2 and 1 of CON1, provide logic-level conversion. This way, the 2.8V signal swings at pins 5 and 6 of IC1 are converted into 5V swings at pins 2 and 1 of CON1, and vice versa. This allows the module to be connected to external circuitry running from a 5V supply, like an Arduino or similar MCU.

The way this kind of ‘passive’ level shifter works is quite clever. Q1 & Q2 are N-channel devices, so they switch on when their gate voltage (“G”) is significantly higher than the source voltage (“S”). At idle, the source is

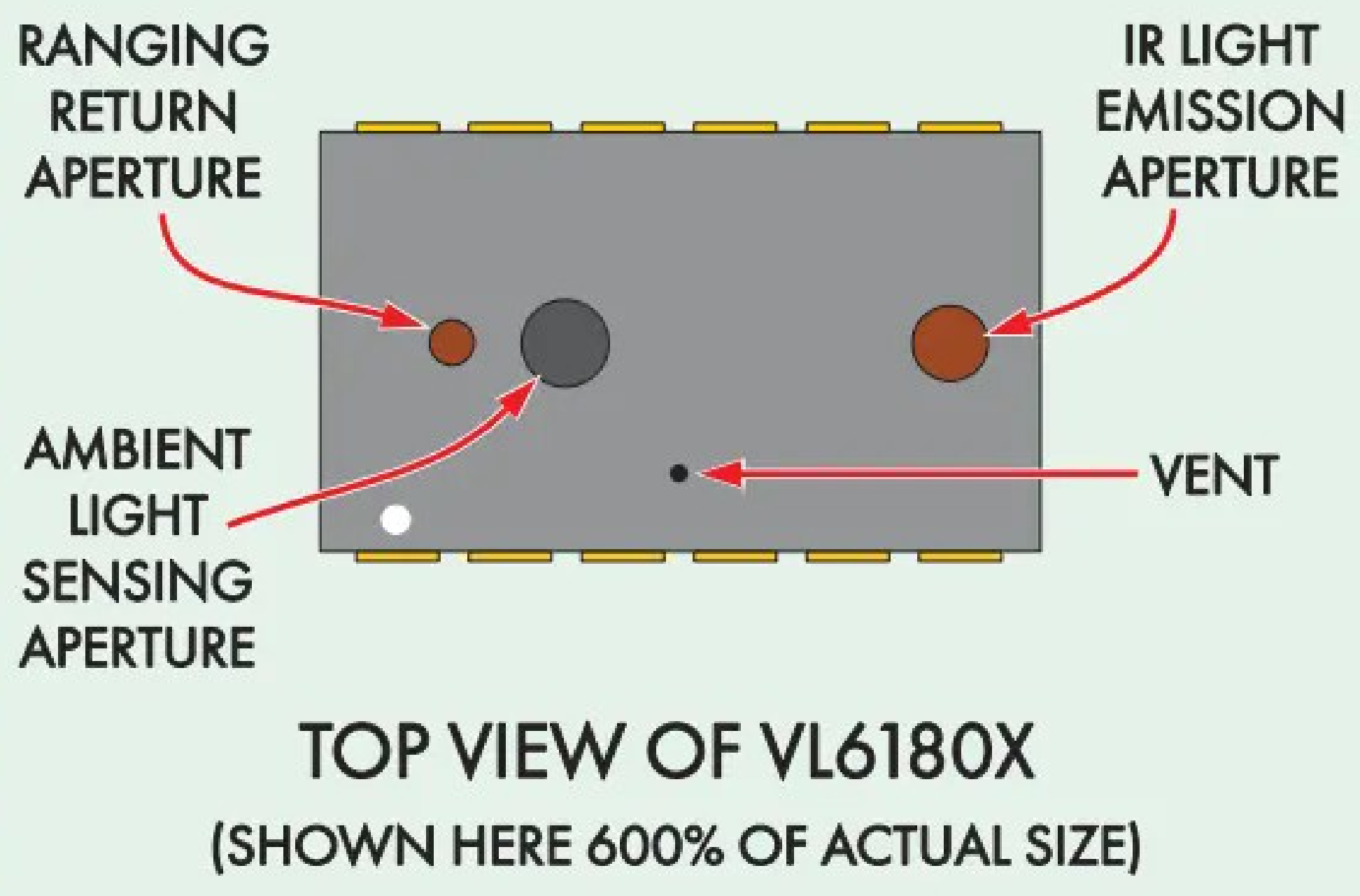

RANGING RETURN APERTURE
IR LIGHT EMISSION APERTURE
AMBIENT LIGHT SENSING APERTURE
VENT
TOP VIEW OF VL6180X
(SHOWN HERE 600% OF ACTUAL SIZE)

Fig.2: the top of the VL6180X IC features three tiny holes that are critical for its functionality. These apertures are required for sensing and emission, with the largest being only 0.58mm in diameter. There is also an even smaller ‘vent’ hole. It’s important to note that the light sensor has a very narrow ‘cone’ and measures objects up to 150-200mm away.

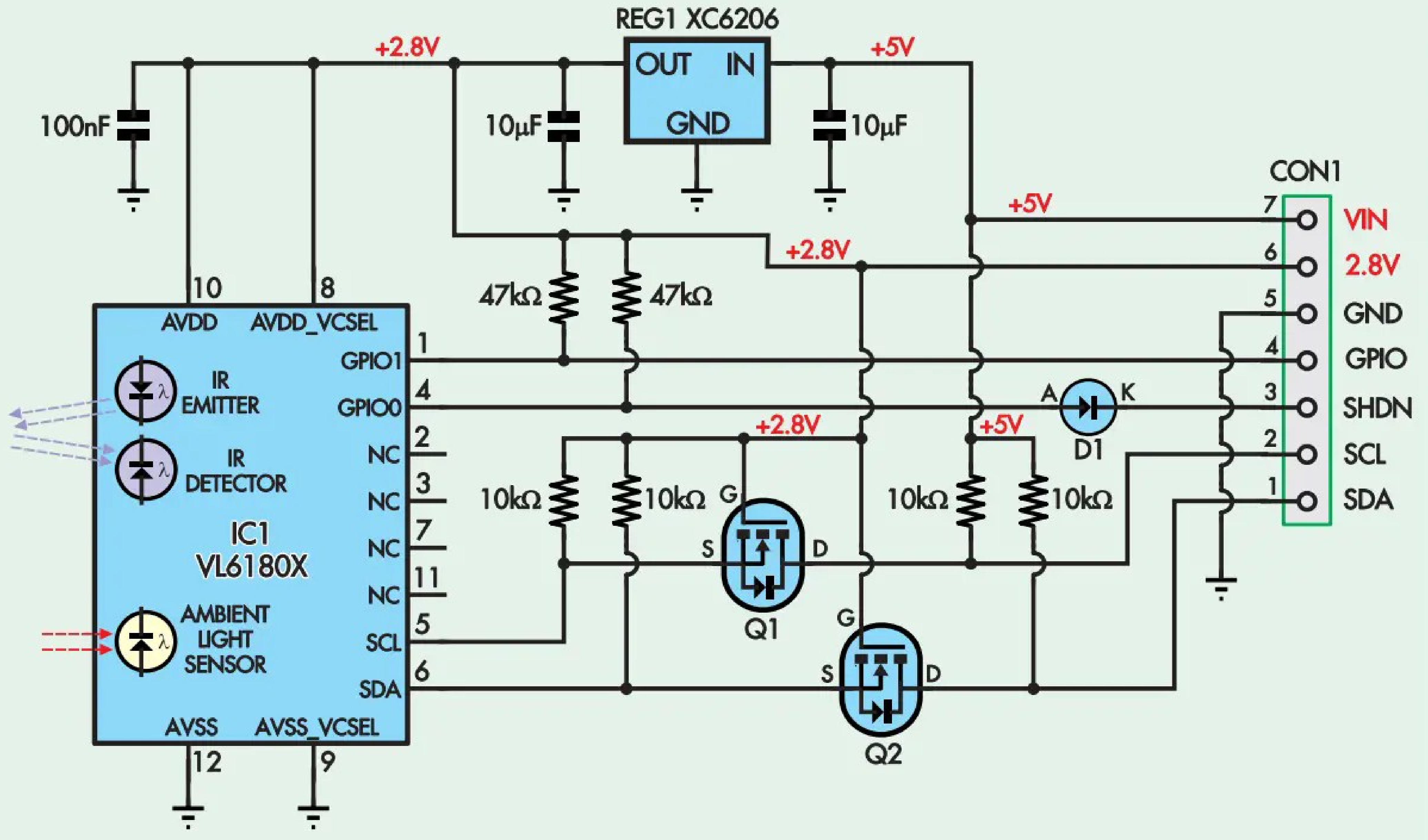

REG1 XC6206
OUT
IN
GND
+2.8V
+5V
100nF
10μF
10μF
CON1
+5V
+2.8V
47kΩ
47kΩ
10kΩ
10kΩ
10kΩ
10kΩ
+2.8V
+5V
10
8
AVDD
AVDD_VCSEL
GPIO1
GPIO0
NC
NC
NC
NC
SCL
SDA
1
4
2
3
7
11
5
6
IR EMITTER
IR DETECTOR
IC1
VL6180X
AMBIENT LIGHT SENSOR
AVSS
AVSS_VCSEL
12
9
A
K
D1
G
S
D
Q1
G
S
D
Q2
7
6
5
4
3
2
1
VIN
2.8V
GND
GPIO
SHDN
SCL
SDA

Fig.3: the circuit diagram for the Quason module which utilises the VL6180X IC. Q1 and Q2 are used for logic-level conversion.

pulled to +2.8V via one 10kΩ resistor, while the drain is pulled to +5V via another. With the gate and source both at +2.8V, the Mosfet is off, so no current flows.

If IC1 pulls its end low, the gate-source voltage becomes +2.8V, so the Mosfet switches on and the corresponding pin on CON1 also goes low. Alternatively, if the pin on CON1 is externally pulled low (eg, by an MCU), the Mosfet is initially off. Still, its parasitic 'body diode' (visible in Fig.3) allows the corresponding pin on IC1 to be pulled down to about +0.7V.

The gate-source voltage of that Mosfet is then 2.8V – 0.7V = 2.1V, high enough for the Mosfet to switch on, pulling the pin on IC1 down to 0V. So when one side goes low, the other does too, but if both sides are allowed to be pulled high by the pullup resistors, they remain high at different voltage levels.

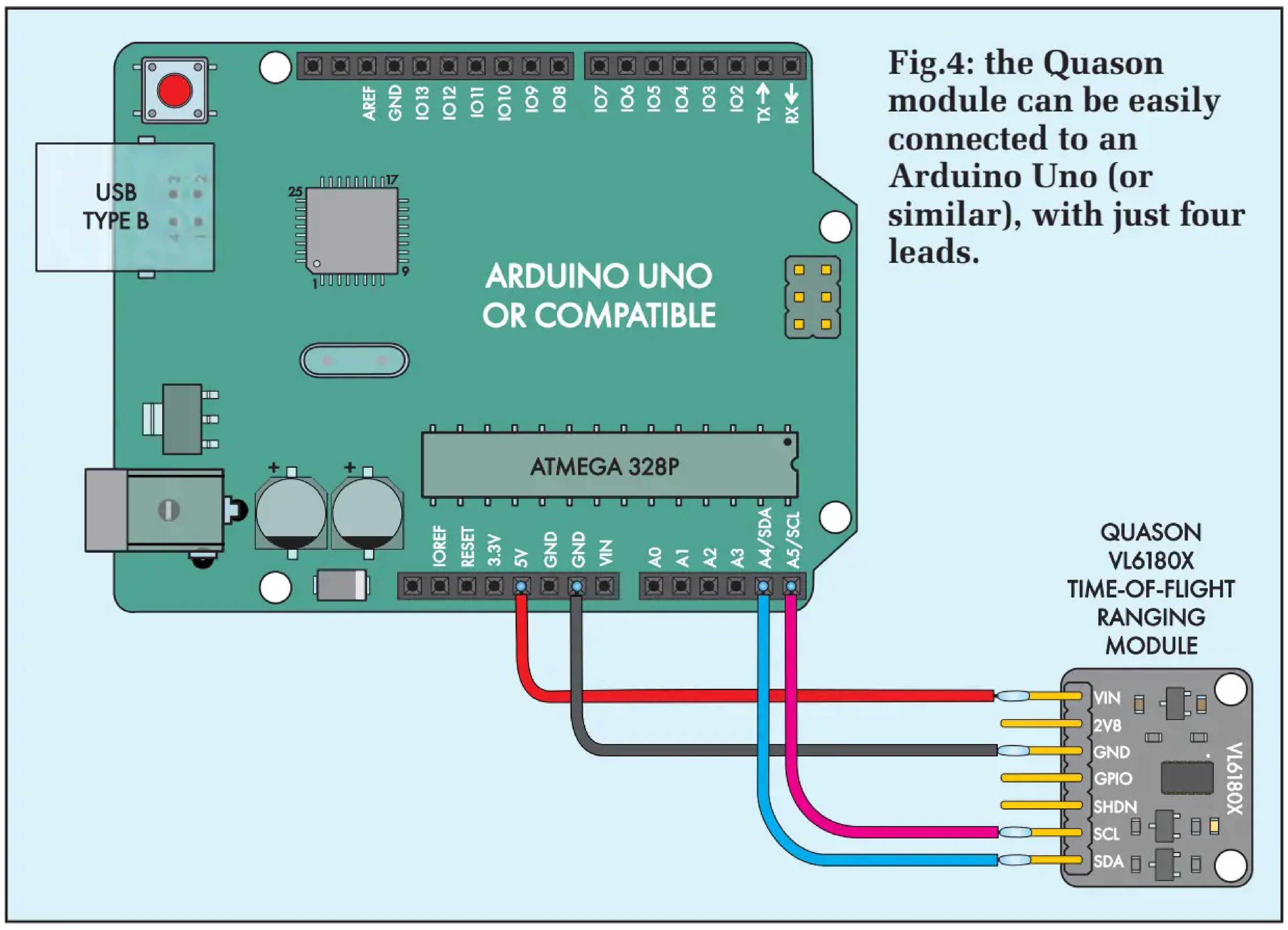


Fig.4: the Quason module can be easily connected to an Arduino Uno (or similar), with just four leads.

Connecting it to an Arduino

As you can see from Fig.4, connecting the module to an Arduino Uno or compatible is very straightforward. The module's VIN pin connects to the Arduino's 5V pin, its GND pin connects to one of the Arduino's GND pins, and its SCL and SDA pins connect to the same pins on the Arduino.

You will also need an Arduino library to get the two communicating, plus a sketch to use the library to make measurements. A couple of these libraries are listed on the Arduino website at www.arduinolibraries.info/libraries/vl6180-x – in both cases, they provide links to the library ZIP files on GitHub.

When you download and unzip either of these libraries, they generously provide example sketches to get you going.

I downloaded one of these libraries, added it to my list of libraries in the Arduino IDE and then loaded one of its example sketches.

It was only a few minutes before I could wave my hand up and down above the VL6180X and see its distance varying in the ranging data on the Arduino IDE's Serial Monitor. It was as simple as that!

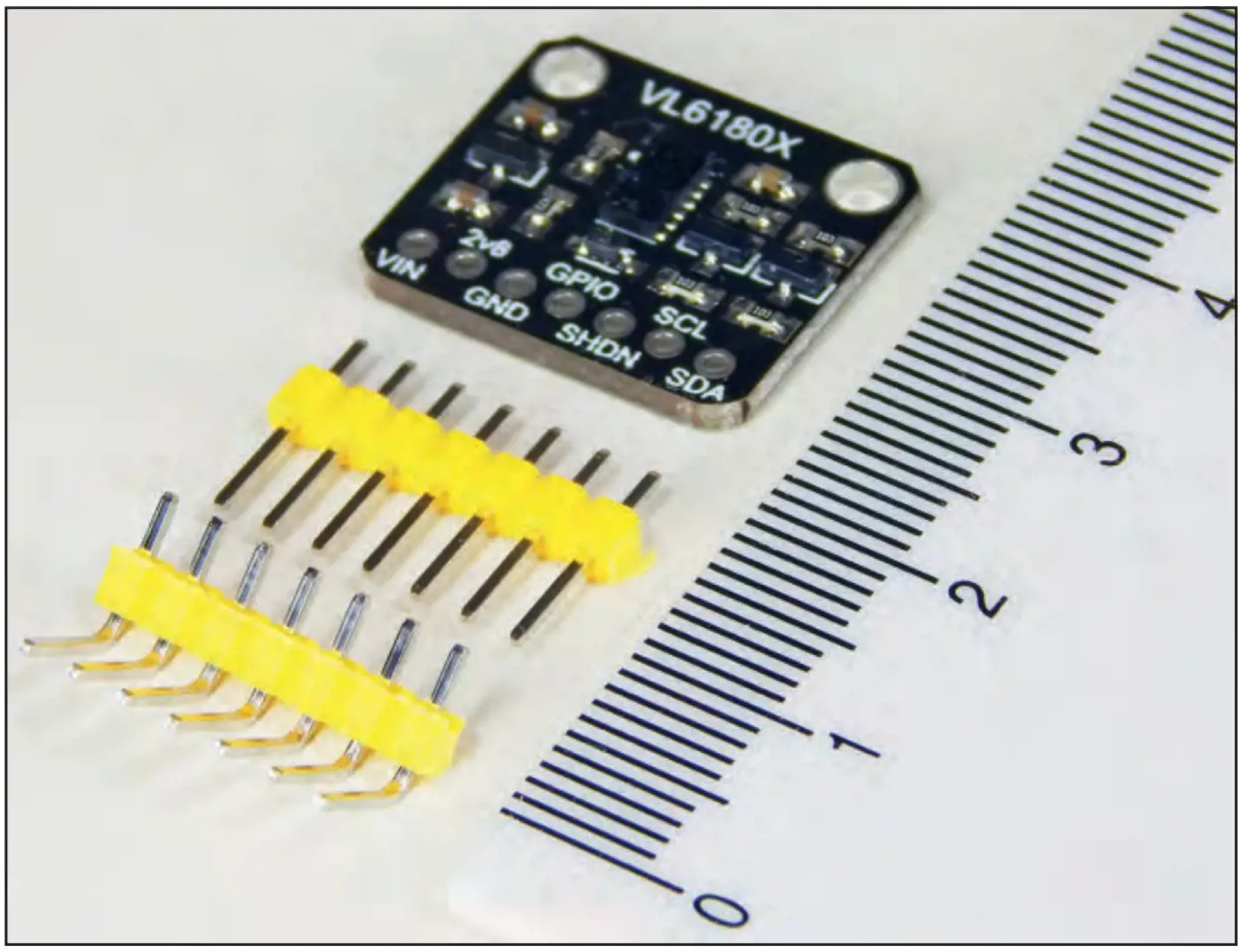
So it's pleasingly easy to get the Quason VL6180X IR range sensing module going with an Arduino. This, plus its low cost, suggests that it would be very suitable for DIY robotics. You might even be able to use a couple of the modules to make a digital Theremin!

Where to get it

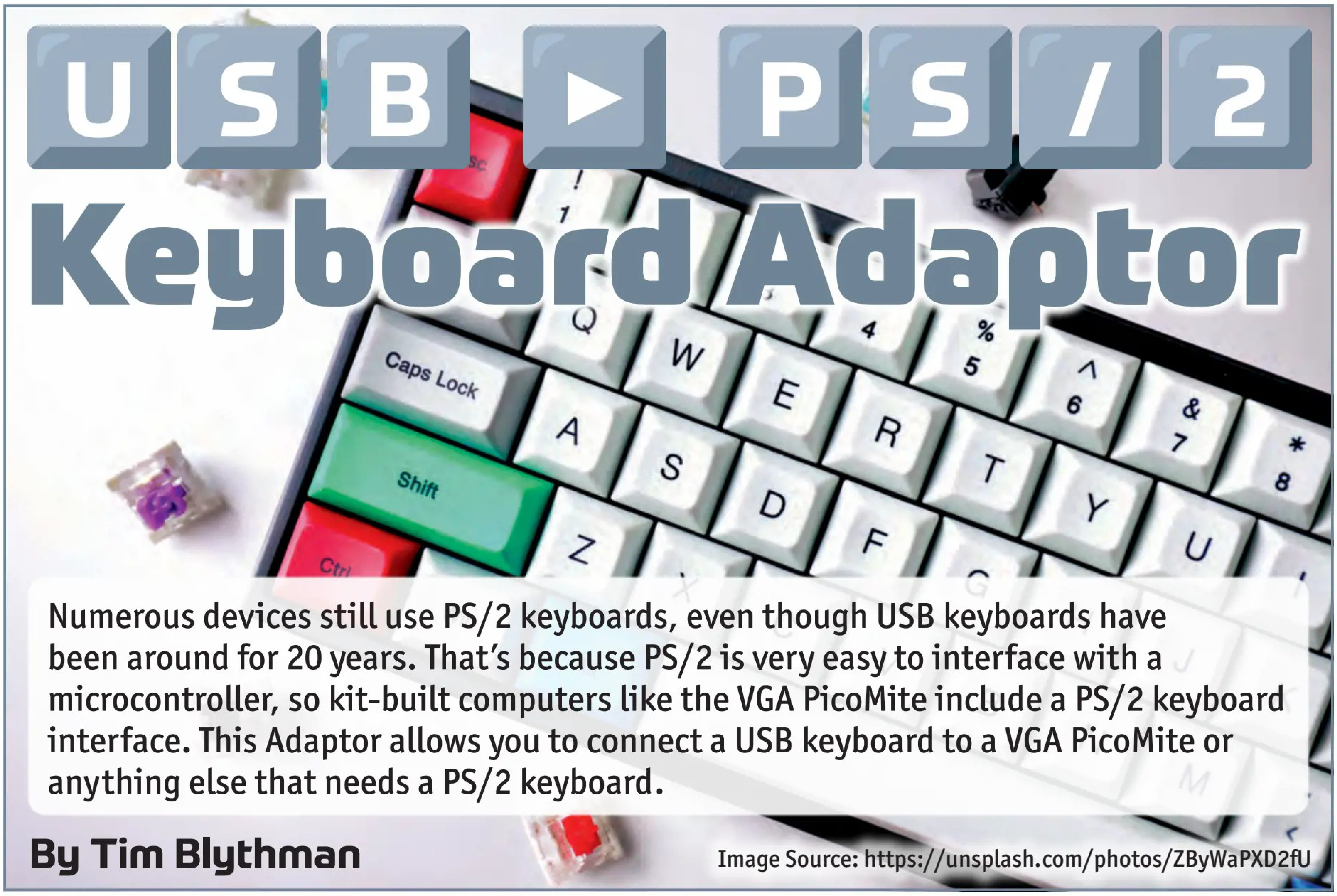
We obtained the module in the photos from the Quason Official Store, one of the vendors on AliExpress (see www.aliexpress.com/item/1005001572022389.html), for £2 including shipping. But there are several other vendors on AliExpress offering it for similar prices, such as SuperModule Store, DIY-Victor Store and HARYE Store.

It is also available from numerous eBay suppliers starting at around £4, including shipping. You could also look on Amazon, which is rapidly becoming another eBay now that they have listings from third-party sellers.

We found a similar (but not identical) module in the online shop https://coolcomponents.co.uk for £8.99, which will presumably be quicker to obtain than those coming from overseas. PE



As you can see from this enlarged photo, the Quason VL6180X is miniature, measuring just 17.8 x 20.3mm.



USB ▶ PS/2 Keyboard Adaptor

Numerous devices still use PS/2 keyboards, even though USB keyboards have been around for 20 years. That's because PS/2 is very easy to interface with a microcontroller, so kit-built computers like the VGA PicoMite include a PS/2 keyboard interface. This Adaptor allows you to connect a USB keyboard to a VGA PicoMite or anything else that needs a PS/2 keyboard.

By Tim Blythman

Image Source: https://unsplash.com/photos/ZByWaPXD2fU

While it's still possible to buy PS/2 keyboards, they aren't as widespread as they used to be. USB keyboards often have better features, and wireless USB keyboards are pretty cheap these days. Wireless PS/2 keyboards exist but are no longer common, making this Adaptor especially useful.

We recently came across a nice design that is based on a Raspberry Pi Pico microcontroller module. It allows a USB keyboard and mouse to connect to a computer that expects a PS/2 keyboard and mouse; see https://github.com/No0ne/ps2x2pico

The hardware presented there is not much more than a Pico board, a level shifter board and some flying leads. We decided to develop a slightly slicker design that would be more robust and easy to use with the VGA PicoMite computer that we published last month.

- VGA PicoMite version kit (SC6861, ~£18)
- ps2x2pico version kit (SC6864, ~£20)
Both kits include everything except the Jiffy box and 6-pin mini-DIN to mini-DIN cable. The mounting hardware and optional headers/sockets are present. The Pico is supplied blank and requires programming.

6-pin mini-DIN cable (SC6869, ~£6): 1.5m long PS/2 cable. Two cables are required if using both the keyboard and mouse.

We're presenting two designs here, with one designed specifically to work with the VGA PicoMite. It should also work fine with any computer system that expects a PS/2 keyboard, including older PCs and boot-to-BASIC computers such as the MaxiMite or Colour MaxiMite.

This project was also prompted by enquiries about the USB Keyboard and Mouse Adaptor for Micros (February 2020), asking us to add a PS/2 interface.

The PS/2 keyboard protocol

"PS/2" refers to the IBM Personal System/2 computer that introduced this interface.

The physical side of the PS/2 protocol is quite simple and consists of two lines that are normally pulled up to 5V. The connected devices (eg, keyboard and host computer) can either leave the lines high or pull them down to 0V using a transistor collector/drain. This allows communications in both directions without conflict.

The lines are called CLOCK and DATA, and electrically, the protocol is very similar to I^2C.

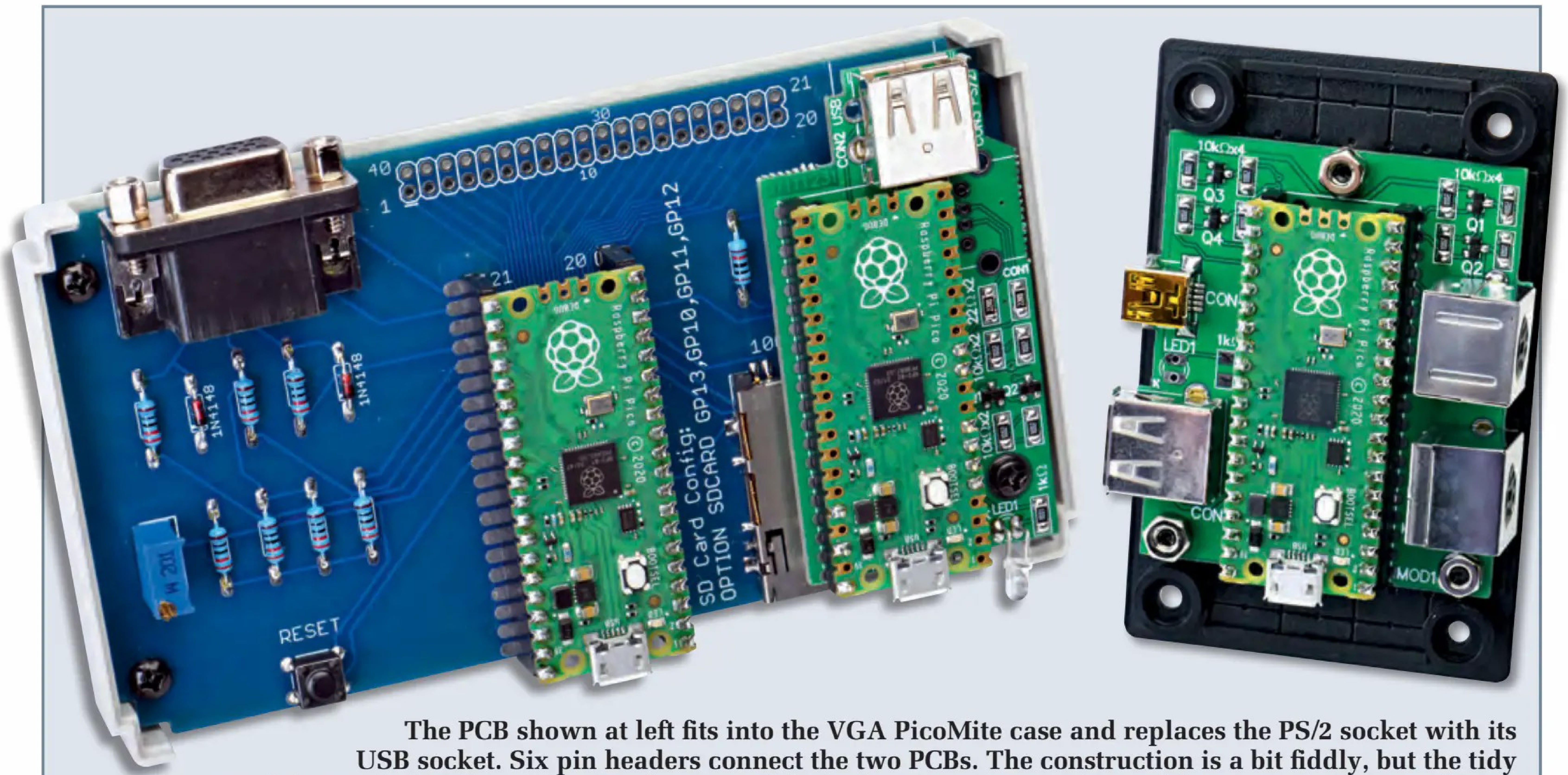
Each byte is sent on the DATA line as a start bit (0), eight bits, an odd parity bit, then a high stop bit. The level on the DATA line is much the same as would be seen on an asynchronous serial line using the '8O1' (eight data bits, odd parity, one stop bit) setting. However, there is a CLOCK line, which is specified as operating at 10-16.7kHz, so it is a synchronous protocol.

The keyboard always provides the CLOCK signal, and the only time the host controls it is to briefly pulse it low to signal that it wants to send data. When the host sends its data, it depends on the keyboard to drive the clock signal as the host drives the DATA line.

Some of the commands the host can send include those to set the lock key LEDs (Caps Lock, Num Lock & Scroll Lock) and to set the 'typematic' rate and delay. Typematic is the term for automatic key press repeats when keys are held down. Since USB keyboards do not implement typematic, we need to emulate that feature.

The host can also query the keyboard

USB to PS/2 Adaptors



The PCB shown at left fits into the VGA PicoMite case and replaces the PS/2 socket with its USB socket. Six pin headers connect the two PCBs. The construction is a bit fiddly, but the tidy result is worth the effort. The PCB shown at right is the ps2x2pico version and has more features such as a mouse adaptor. Due to its larger size, it needs to be fit into a UB5 Jiffy box, but it can convert a USB keyboard/mouse combo to work with two PS/2 ports and it supports wireless USB devices.

about its status and identity. For the most part, though, data is sent from the keyboard to the host when keys are pressed or released. The PS/2 'scancodes' that the keyboard sends do not neatly map to anything like the ASCII codes or the USB scancodes sent by USB keyboards.

In fact, PS/2 scancodes correspond to the original physical location of the keys on the keyboard, as adjacent keys often have similar codes. The mapping was clearly chosen to simplify the scanning and encoding of keys by the keyboard, leaving the hard work of decoding the scancodes to software on the host computer.

A PS/2 mouse works similarly, although it sends button presses and movement changes instead of scancodes. The host can command it to set parameters like sampling (update) rate and scaling.

So, apart from the scancode interpretation that is needed, the PS/2 protocol is fairly simple and is easy enough to implement as it is driven by the keyboard. We can control the clock rate since we are trying to emulate a keyboard.

Circuit details

As mentioned earlier, the first circuit is specifically designed to work with the VGA PicoMite. A second circuit is intended to match the circuit used by No0ne's ps2x2pico software. PCBs are available for both.

Electrically, both are very simple, and most components are present to interface a 3.3V Pico microcontroller board to the 5V levels used by the PS/2 interface.

Each circuit has a corresponding PCB; we will describe their differing software and construction later. We think the first version (Fig.1) is the best choice if you want to connect a USB keyboard to a VGA PicoMite. However, if you're going to interface to an old PC with both PS/2 mouse and keyboard connections, we recommend building the second version.

VGA PicoMite version

Fig.1 shows the circuit for this version of the Adaptor. It uses a Raspberry Pi Pico microcontroller board (MOD1), a pair of USB connectors (CON1 and CON2), a 6-pin mini-DIN socket (as used for PS/2), CON3, and a few other components.

USB sockets CON1 and CON2 are connected in parallel. Those who know USB will realise that both connectors cannot be used simultaneously; they are alternatives, and only one should

The Adaptor is a neat install in a VGA Picomite, replacing the PS/2 socket with a USB socket.

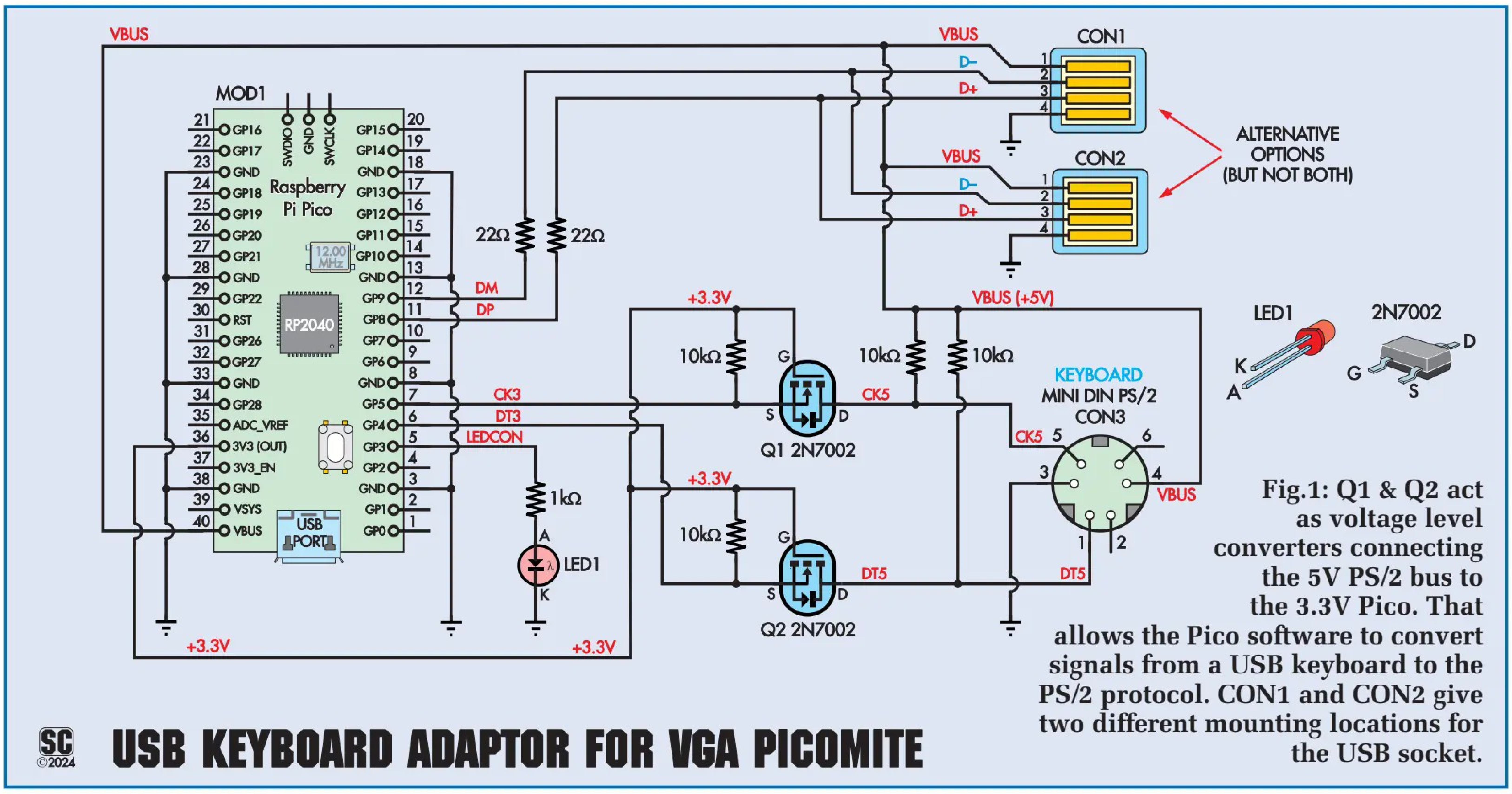

VBUS
MOD1
Raspberry Pi Pico
RP2040
12.000 MHz
USB PORT
SWDIO
GND
SWCLK
GP16
GP17
GND
GP18
GP19
GP20
GP21
GND
GP22
RST
GP26
GP27
GND
GP28
ADC_VREF
3V3 (OUT)
3V3_EN
GND
VSYS
VBUS
GP15
GP14
GND
GP13
GP12
GP11
GP10
GND
GP9
GP8
GP7
GP6
GND
GP5
GP4
GP3
GP2
GND
GP1
GP0
22Ω
22Ω
DM
DP
CK3
DT3
LEDCON
1kΩ
LED1
A
K
+3.3V
+3.3V
+3.3V
+3.3V
10kΩ
10kΩ
10kΩ
10kΩ
G
S
D
Q1 2N7002
Q2 2N7002
CK5
DT5
VBUS (+5V)
VBUS
D–
D+
CON1
CON2
ALTERNATIVE OPTIONS (BUT NOT BOTH)
KEYBOARD MINI DIN PS/2 CON3
CK5
DT5
VBUS
LED1
2N7002
K
A
G
D
S
SC ©2024
USB KEYBOARD ADAPTOR FOR VGA PICOMITE

Fig.1: Q1 & Q2 act as voltage level converters connecting the 5V PS/2 bus to the 3.3V Pico. That allows the Pico software to convert signals from a USB keyboard to the PS/2 protocol. CON1 and CON2 give two different mounting locations for the USB socket.

be fitted. They are the same socket type (both type A) but are in different locations on the PCB to suit various applications.

The data lines from CON1 (or CON2) head via 22Ω resistors to consecutive pins on MOD1. Although the RP2040 chip on the Pico has native support for USB, some clever people have written a library that uses the Pico's PIO (programmable input/output) peripheral as a USB controller. The general-purpose I/O pins do not have internal 22Ω resistors as required for USB communications, hence our adding them.

Two more of the Pico's pins (carrying the CLOCK and DATA signals) connect to a level-shifting arrangement based on 10kΩ resistors and Mosfets Q1 and Q2. An identical arrangement is used on the VGA PicoMite to interface the 3.3V Pico to the 5V levels used on the PS/2 bus.

This configuration is well-suited to voltage level conversion on open-collector busses and is commonly used with I^2C interfaces. The resistors pull up the lines on each side to either 3.3V or 5V. The gate is at the same voltage as the source, so the Mosfet's channel is off, and its body diode is reverse-biased due to the 5V rail being higher than the 3.3V rail.

If the 3.3V logic line (connected to the Mosfet source) is pulled down, the gate is at a higher voltage than the source and the Mosfet switches on, propagating the low level to the 5V logic side. If the 5V logic line goes low, the source is pulled down via the Mosfet's body diode, the Mosfet switches on, and the low level is also seen on the 3.3V side.

Releasing the low level allows the source to rise until the Mosfet switches off and each side returns to its initial state, with both sides pulled up by their respective resistors.

One more pin of the Pico is connected to a 1kΩ resistor and then to ground via the LED. The latter is a status indicator, with the LED lit by bringing the digital pin high.

The 5V pins of all the connectors are tied together so that any connector can supply 5V as needed. Since CON1 (or CON2) is a host USB-A port, it will power a downstream USB device like a keyboard.

In normal operation, power will come from CON3, since it will be connected to a PS/2 host. If necessary, power could be provided to the circuit via the Pico's USB socket. The Pico has an integrated 3.3V regulator, with its output available at the 3V3 OUT pin. In this case, it is only used as a reference voltage for the level-shifting circuitry.

The ps2x2pico version

Fig.2 shows the second circuit. It has a Pico (MOD1), USB type-A socket (CON1), two 6-pin mini-DIN sockets (CON2 & CON3) and a mini-USB socket (CON4). CON4 only has its power (VBUS & GND) pins connected.

The ps2x2pico name has been coined for the software by its creator, No0ne. We are simply using it to identify the version of the hardware that we have developed to work with their software.

The eight 10kΩ resistors and four Mosfets implement four logic level-shifting channels identically to the first circuit. The 5V sides of two channels go to CON2 for a PS/2 keyboard, while the other two connect to CON3 for a PS/2 mouse.

The 3.3V sides of the level shifters connect to pins on MOD1, while CON1 connects the USB data lines to a pair of pads. These are intended to be connected to a matching pair of pads on the underside of the Pico, and thus the USB D– and D+ lines on its USB controller.

Finally, LED1 and its 1kΩ ballast resistor connect to another of the Pico's I/O pins.

It's possible to replace CON1 with a USB-OTG adaptor fitted into the Pico's micro-USB socket; that is what is shown in the photos at https://github.com/No0ne/ps2x2pico

We felt that having fixed sockets made for a more robust solution. The fixed socket arrangements also lend our final PCB design to being installed in a Jiffy box.

Software

Despite their almost identical functionality, the two circuits use vastly

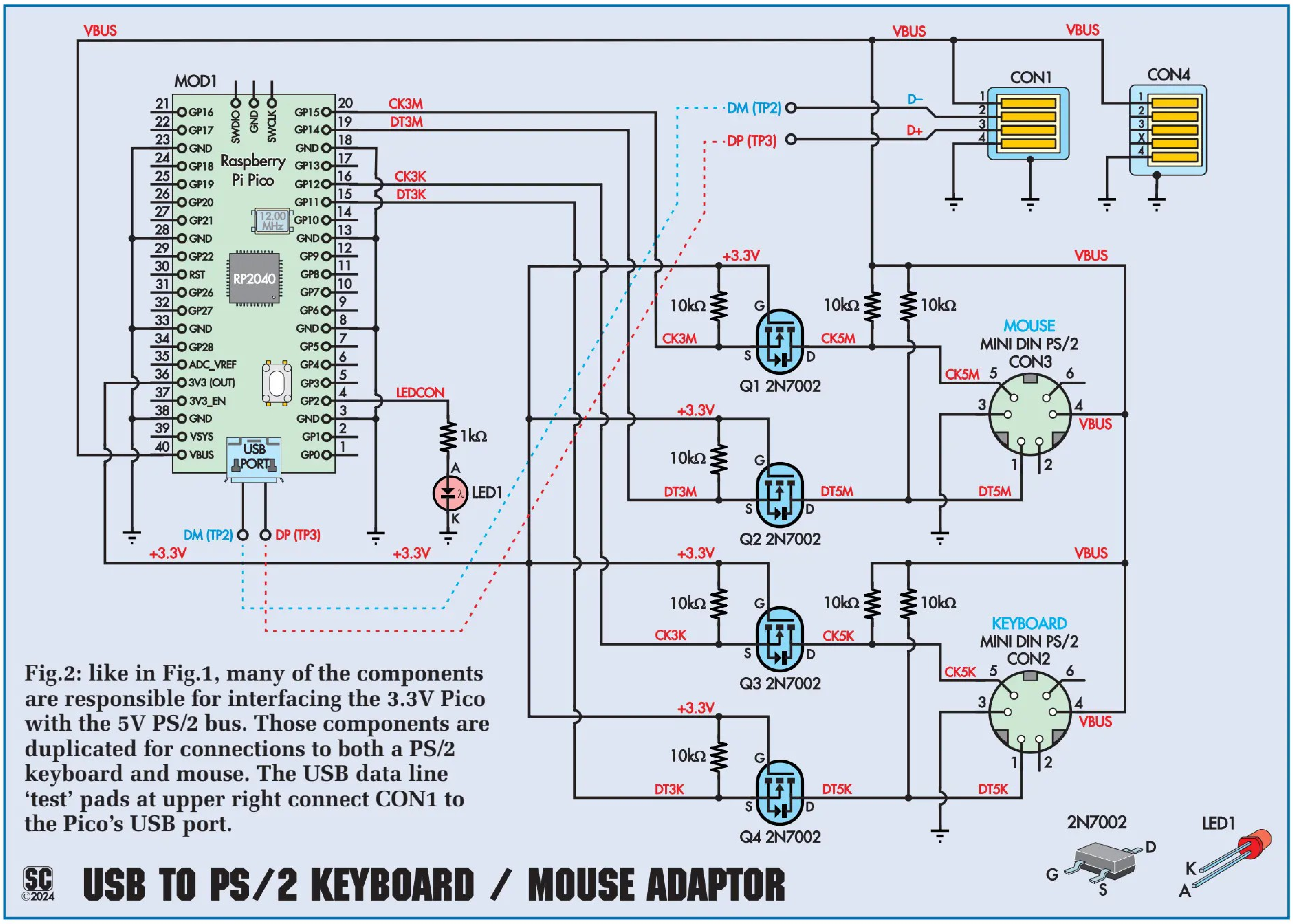

VBUS
MOD1
Raspberry Pi Pico
RP2040
12.00 MHz
USB PORT
SWDIO
GND
SWCLK
GP16
GP17
GND
GP18
GP19
GP20
GP21
GND
GP22
RST
GP26
GP27
GND
GP28
ADC_VREF
3V3 (OUT)
3V3_EN
GND
VSYS
VBUS
GP15
GP14
GND
GP13
GP12
GP11
GP10
GND
GP9
GP8
GP7
GP6
GND
GP5
GP4
GP3
GP2
GND
GP1
GP0
CK3M
DT3M
CK3K
DT3K
LEDCON
1kΩ
LED1
A
K
DM (TP2)
DP (TP3)
+3.3V
D–
D+
CON1
CON4
10kΩ
G
S
D
CK5M
DT5M
CK5K
DT5K
Q1 2N7002
Q2 2N7002
Q3 2N7002
Q4 2N7002
MOUSE
MINI DIN PS/2
CON3
KEYBOARD
MINI DIN PS/2
CON2
2N7002
SC ©2024

Fig.2: like in Fig.1, many of the components are responsible for interfacing the 3.3V Pico with the 5V PS/2 bus. Those components are duplicated for connections to both a PS/2 keyboard and mouse. The USB data line 'test' pads at upper right connect CON1 to the Pico's USB port.

USB TO PS/2 KEYBOARD / MOUSE ADAPTOR

different software implementations. The code at https://github.com/No0ne/ps2x2pico is built using the Pico's C SDK (software development kit). It uses the RP2040's internal USB controller peripheral in host mode, and implements the PS/2 interface using the PIO peripheral.

The PIO is a programmable state machine that can be used to emulate many I/O and communications peripherals. It was used in the VGA PicoMite project published in the December 2024 issue to generate the VGA video output signals.

Using the internal USB controller in host mode is easier but it also means that the controller cannot operate in device mode, for example, to provide debugging data over a virtual serial port. Plus the USB data connection must be made via the test pads on the Pico, instead of standard header pins.

Our software instead uses the PIO to emulate a USB host peripheral based on a library available within the Arduino IDE. Thus, we used the Arduino IDE to build our software.

The USB host implementation means that the PIO peripheral cannot provide the PS/2 interface, so we have written it to work using GPIO pins and timer interrupts instead. This timer operates at 50kHz and is divided into four phases to give a nominal 12.5kHz PS/2 clock frequency.

Using the Arduino IDE also allows us to customise the code more easily, and we have ensured that it works well with the VGA PicoMite.

Some key events require more than one byte to be sent on the PS/2 line, so a queue has been implemented to ensure that data moves in an orderly fashion. It should also guard against brief bursts of keyboard activity overwhelming the Adaptor.

As we mentioned, the software must map scancodes from the USB scanset to the PS/2 scanset. We use what is known as Set 2, the default for PS/2 keyboards.

The mapping is not quite one-to-one. USB keyboards report the state of the modifier keys (Ctrl, Shift, Alt etc) as bits in a status byte rather than as scancode events. So, we have to convert the changes in these status bits into the key-up and key-down events that PS/2 keyboards generate.

Some keys, such as Pause, have odd mappings that must be handled uniquely. That is because the Pause feature was originally invoked by the Ctrl+NumLock key combination, meaning that a single keystroke maps to eight bytes to send on the PS/2 line.

The Adaptor must also send repeated key-down events to emulate the typematic feature. With USB keyboards, that is usually handled by the host computer's software.

The ps2x2pico software does not control the LED provided on the second circuit; we simply added it to the PCB in case users wish to modify or update the software to do so.

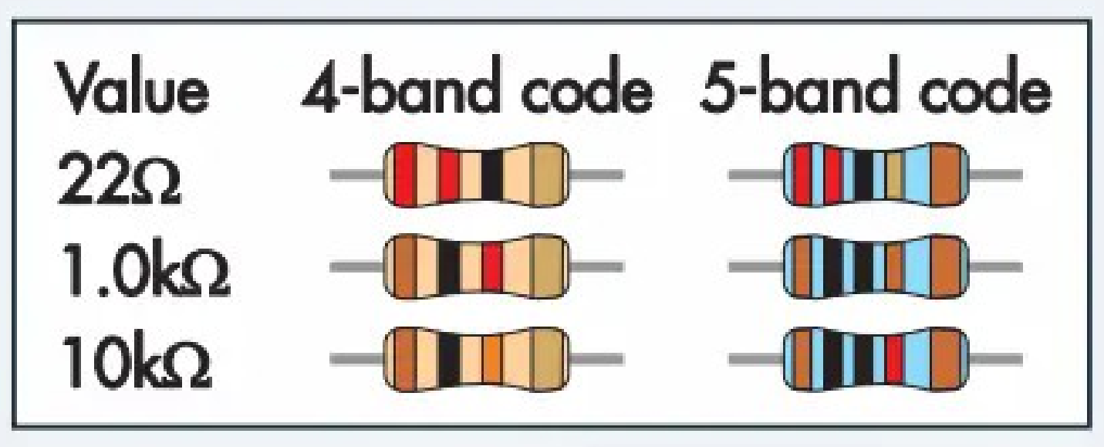
Note that both versions of the software support USB hubs, so if you need to attach a separate keyboard and mouse, you only need to add a hub. If you are using a wireless keyboard and mouse, consider buying them together; in that case, both keyboard and mouse will usually share a single wireless USB receiver.

Parts List – VGA PicoMite USB to PS/2 Converter

1 PCB double-sided PCB coded 07111231, 42 × 66mm
1 Raspberry Pi Pico programmed with 0711123A.UF2 (MOD1)
1 UB5 Jiffy box ●
2 M3 × 10mm panhead machine screws ●
4 M3 hex nuts ●
2 M3 flat washers ●
6 header pins ♦
1 M3 × 5mm panhead machine screw ♦
1 M3 × 6mm tapped spacer ♦
2 20-way pin headers (**optional**, for MOD1)
2 20-way female header sockets (optional, for MOD1)
1 PCB mount USB-A horizontal socket (CON1 or CON2)
1 6-pin mini-DIN socket (CON3) ●
1 6-pin mini-DIN to 6-pin mini-DIN cable ●

Semiconductors
1 3mm green LED (LED1)
2 2N7002 N-channel Mosfets, SOT-23 (Q1, Q2)

Resistors *(all M3216/1206 SMD, ¼W)*
4 10kΩ 1 1kΩ 2 22Ω



Value	4-band code	5-band code
22Ω		
1.0kΩ		
10kΩ		

● *only needed if installing the Adaptor in a Jiffy box*
♦ *only required if installing the Adaptor inside a VGA PicoMite*

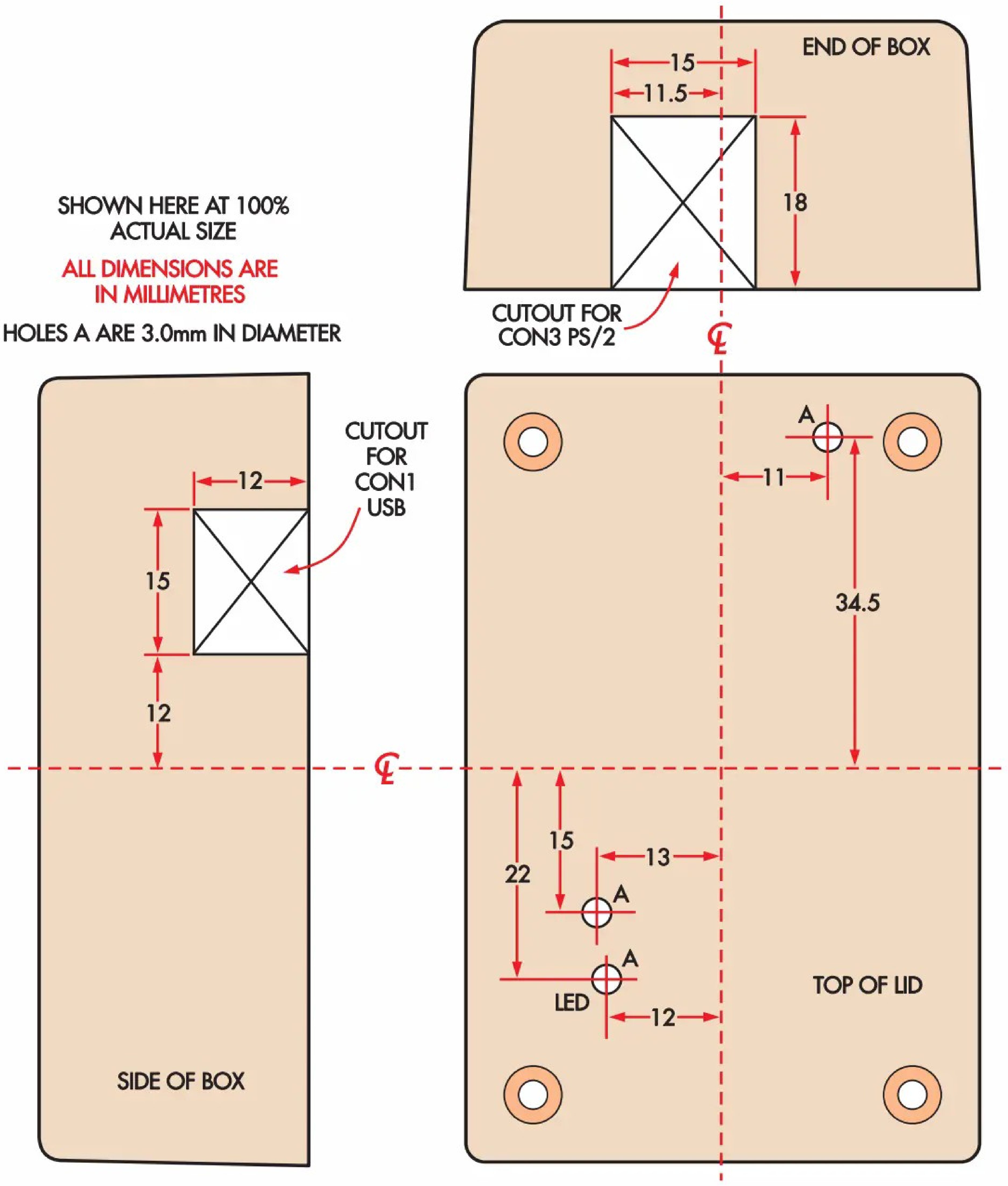


Fig.3: three 3mm holes are needed in the Jiffy box lid to accommodate the PCB coded 07111231. Two are for mounting screws and the third allows the LED to shine through. You could use the blank PCB as a jig to mark out the holes or confirm your measurements.

Hardware

The VGA PicoMite version PCB has been designed so it can sit directly above the main PCB of the VGA PicoMite and fit into the VGA PicoMite's recommended case, as shown on page 43. In this configuration, the CON3 PS/2 connector is not fitted, and the corresponding pads on the two PCBs are directly connected with header pins or similar (shown on page 50).

There is the option of fitting the USB socket where the PS/2 socket would have been, meaning that you can now plug a USB keyboard in where you would have otherwise plugged a PS/2 keyboard.

You could use the other USB socket location instead, although you would need to cut a hole in the side of the VGA PicoMite case to access it.

Another option is to mount either PCB inside a UB5 Jiffy box. The assembled PCB is secured to the Jiffy box's lid. Slots for the various connectors can be made by simply cutting down from the top edge of the box, which is easier than trying to hollow out a shape in the side of the box.

In this configuration, the LED should be mounted on the underside of the PCB to allow it to shine through the box lid. We'll provide more guidance on these options later.

Programming the Pico

We recommend programming the Pico before fitting it to the board, especially since the ps2x2pico version uses the USB socket normally used for program uploads.

Connect the Pico to a computer, holding in the white BOOTSEL button as you do so. A drive labelled RPI-RP2 should appear. Upload the firmware by copying the respective UF2 file to that drive.

Use the file "0711123A.UF2" for the VGA PicoMite version. After programming, it will reappear as a virtual USB-serial port, so you can check that such a device appears on your system. Sending a ~ character to that serial port will toggle debugging mode, but you won't see much of note until it is connected in-circuit.

The ps2x2pico version requires the "ps2x2pico.UF2" file. There won't be any obvious clues that programming has completed except that the drive will disappear. The ps2x2pico version behaves as a USB host, so you shouldn't see any USB devices.

That software is under active development, so keep an eye out for updates. We used version 0.7 in our testing and it is included in the software downloads. This software is copyrighted by No0ne and released under an MIT open-source license.

In the software bundle, we've also provided a PS2_HOST sketch that we used for testing. The bundle includes the UF2 file for this sketch. It is designed to work with the VGA PicoMite version of the hardware, and simply provides a PS/2 host port on the CON3 mini-DIN socket. Communication is via a virtual USB serial port on the Pico (via the micro-USB socket).

This software will report any packets received and their equivalent keys (if the device is a keyboard). Host packets can be sent by typing their hexadecimal codes followed by Enter. There is an assortment of PS/2-related links at the bottom of https://github.com/No0ne/ps2x2pico and some of those list host commands.

Construction

The first circuit, the VGA PicoMite version, corresponds to the PCB coded 07111231 (34 × 65.5mm). The second version that supports both a keyboard and mouse uses a PCB coded 07111232 (48 × 58mm); we will refer to it as the ps2x2pico version.

The Keyboard Adaptor can be fitted inside the VGA PicoMite case or a Jiffy box as shown here. This enclosure has the holes made as per Fig.3.

To keep everything compact, we've primarily used surface-mounting components. You should have tools such as a fine-tipped iron, tweezers and magnifiers. Useful consumables include solder wire, flux paste and solder-wicking braid.

The small PCBs can be held in place with Blu-Tack or similar while soldering (if you don't have a PCB-holding vice). Having an appropriate solvent on hand is also a good idea, so you can clean up any flux residue left after soldering.

If you are building a version to fit into a Jiffy box, you can use the bare PCB to mark out the location of the holes that are needed. That is usually easier than using our drilling and cutting diagrams, although you have that option too.

There are two holes for M3 mounting screws in the VGA PicoMite version, plus a 3mm hole for the LED, as shown in Fig.3. The ps2x2pico version uses three mounting holes, plus one LED hole, although the default software does not make use of the LED, so you could omit it. Fig.4 is the cutting and drilling diagram for that version.

The screw holes can easily be marked by running a pencil (the thin tip of a 0.7mm mechanical pencil is ideal) around or through the holes. The LED holes can be marked using the two holes through their pads on the PCB. Draw a line between those marks and then a smaller line across the exact middle of the one you drew to find the centre of the LED hole.

There are some cuts to be made in the sides of the boxes, which are easily made by using the assembled PCBs as guides while mounted on the lids.

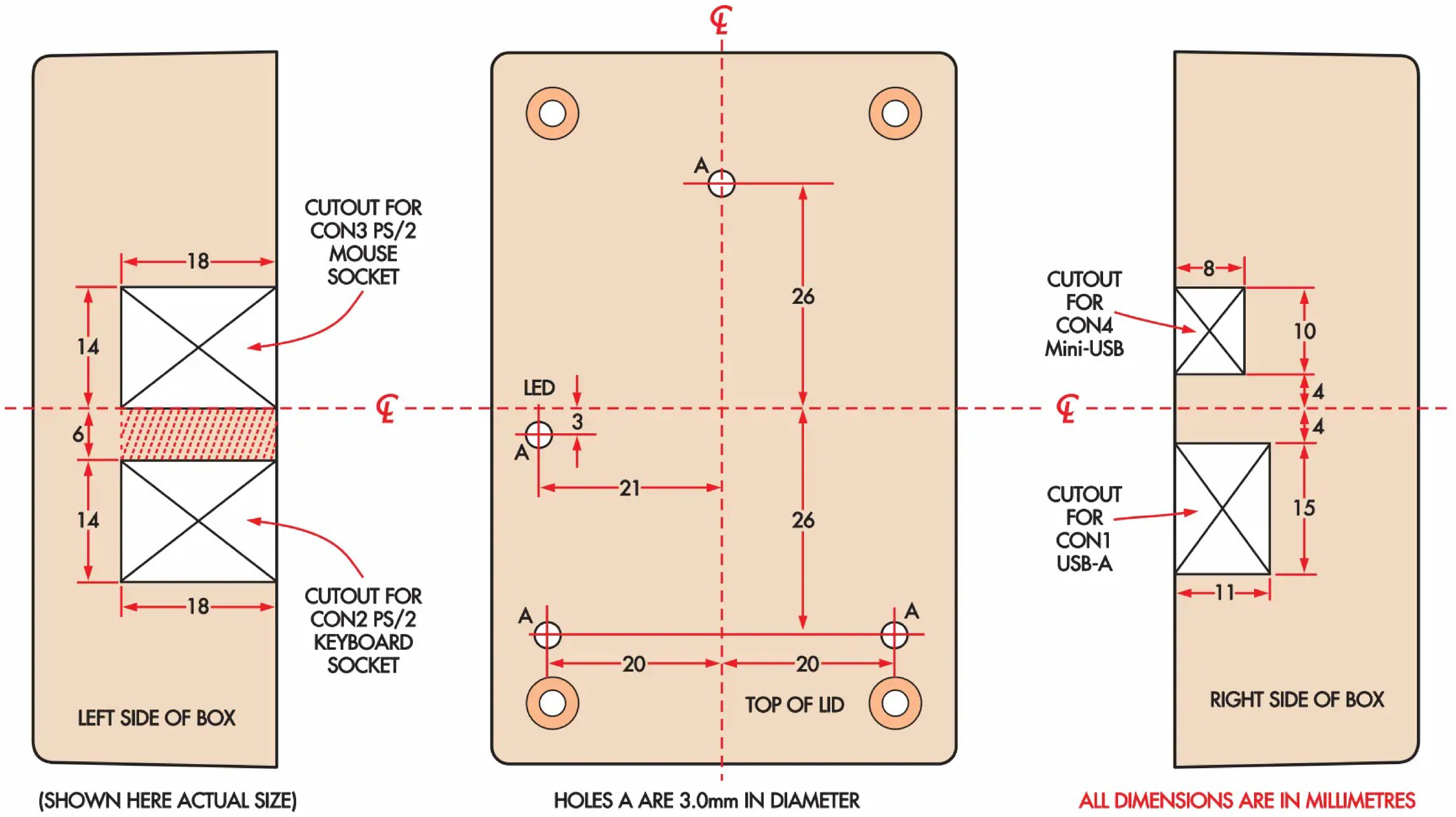


Fig.4: the hole at centre left is only needed if the LED is fitted; the other three holes are for mounting the PCB coded 07111232. The cutout regions are necessary to accommodate the various sockets. You might find it easier to remove the red shaded region and glue the tab back in later.

VGA PicoMite ver. assembly

The PCB overlay for this version is shown in Fig.5. Use that and the photo shown opposite as a guide to fitting the parts to the PCB.

Before assembling the PCB, if you are going to install it in the VGA PicoMite, carefully break off the two PCB tabs that are attached by ‘mouse bites’. Grasp the main part of the PCB with one hand, then use a pair of wide-nosed pliers to flex and break off the tabs.

Do this outside to avoid inhaling any fragments of fibreglass. If you need to use a file to clean up the rough edges of the mouse bites, do that outside too.

If you are fitting this PCB to a Jiffy box, one tab is used to help secure the PCB while the other fills the gap in the edge of the box adjacent to the USB socket. Neither is essential to the electrical operation of the circuit.

Apply flux to the PCB for the surface mounting components. Place each in their marked locations, using the relevant overlay diagram as a guide, and tack one lead. Check that each component is within their pads before soldering the remaining lead(s).

Use your solvent to clean up the PCB after the surface mounting components have been fitted and allow the PCB to dry.

Solder LED1 next. There are cathode (K) markings on both sides of the PCB to allow it to be fitted to either side. If you are installing it in the Jiffy box, solder it to the back of the PCB (the side with the Silicon Chip logo). It should be flush against the PCB.

If you are fitting it to the VGA PicoMite then solder it to the front of the PCB, with about 10mm of lead between the LED’s body and the PCB. Later, the LED’s leads can be bent to aim it out through the front of the enclosure near the SD card socket. Of course, you should ensure the shorter lead goes to the pin marked K.

Now connect the Pico to the PCB; there are three ways to do this. Firstly, you could solder the Pico directly to the PCB. This requires no extra parts but means that the Pico needs to be accurately aligned to the pads on the PCB. You could use M2 screws in the corner holes to temporarily secure the Pico, ensuring correct alignment during soldering.

Or you could instead solder pin headers to the Pico (as though you were going to use it on a breadboard). You could then solder female header sockets to the PCB and slot the Pico onto them, or solder the Pico’s headers directly to the PCB.

If installing the Adaptor in a Jiffy Box, there are no height requirements that would restrict using headers. The clearance is tighter inside the VGA PicoMite case, so you need to solder the Pico directly to the PCB or use male headers only; using sockets makes the assembly too tall.

ps2x2pico ver. construction

Fig.6 is the PCB overlay for this version. You can use that and the photo shown next to Fig.6 to help you assemble the PCB.

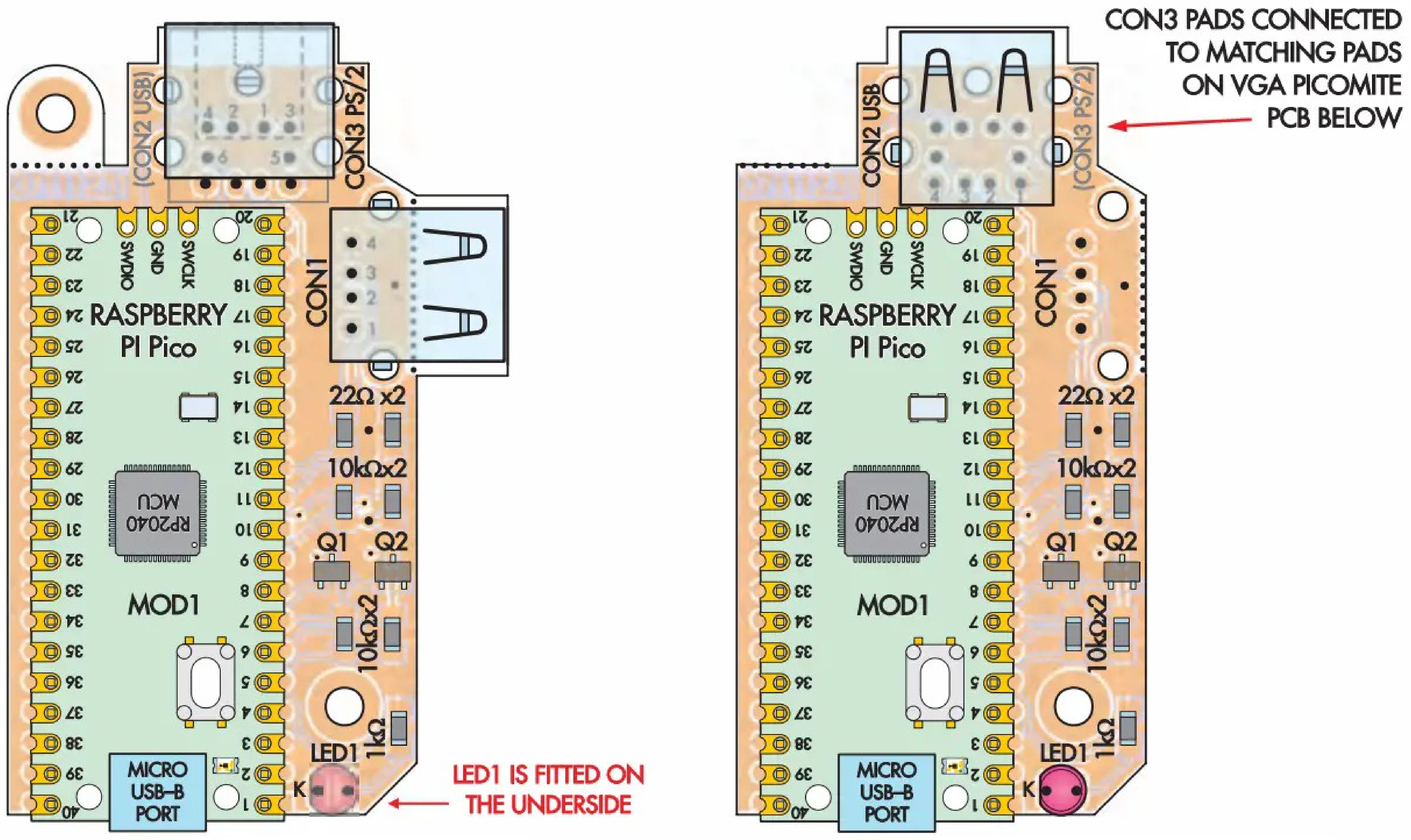
To fit out the PCB for the ps2x2pico version, apply flux to the pads for all the surface-mounting components, including CON4, the mini-USB socket. If you only plan to use the default firmware, you can omit the 1kΩ resistor and the LED, as that firmware will not drive them.

If you want to modify the firmware to use the LED, it should be fitted to the rear of the PCB if you are using the Jiffy case. Otherwise, you might like to fit it to the top if you are using the bare board.

Place CON4 first and locate it on the PCB with the pegs on its underside. Clean the iron’s tip and apply a small amount of fresh solder. Touch it to the pad on the PCB and the solder should run onto the lead. Use a generous amount of heat and solder to secure the four larger pads that connect to the shell and use the solder wick to draw away any excess solder if there are bridges between the pads.

Then fit the four transistors, followed by the eight resistors. With the surface-mounted parts in place, you can clean off the excess flux and allow the PCB to dry.

As with the VGA PicoMite version, you can install the Pico directly to the PCB or on headers. An important difference is that the two small pads (TP2 and TP3) on the Pico near its micro-USB socket must also be connected to



WHEN ADAPTOR IS FITTED INTO A JIFFY BOX

WHEN ADAPTOR IS ATTACHED TO A VGA PICOMITE

Fig.5: assembly of the PCB designed for the VGA PicoMite is straightforward, as there are only 13 components. It’s important to make sure the orientation of the LED and the Pico is correct when attaching those components.

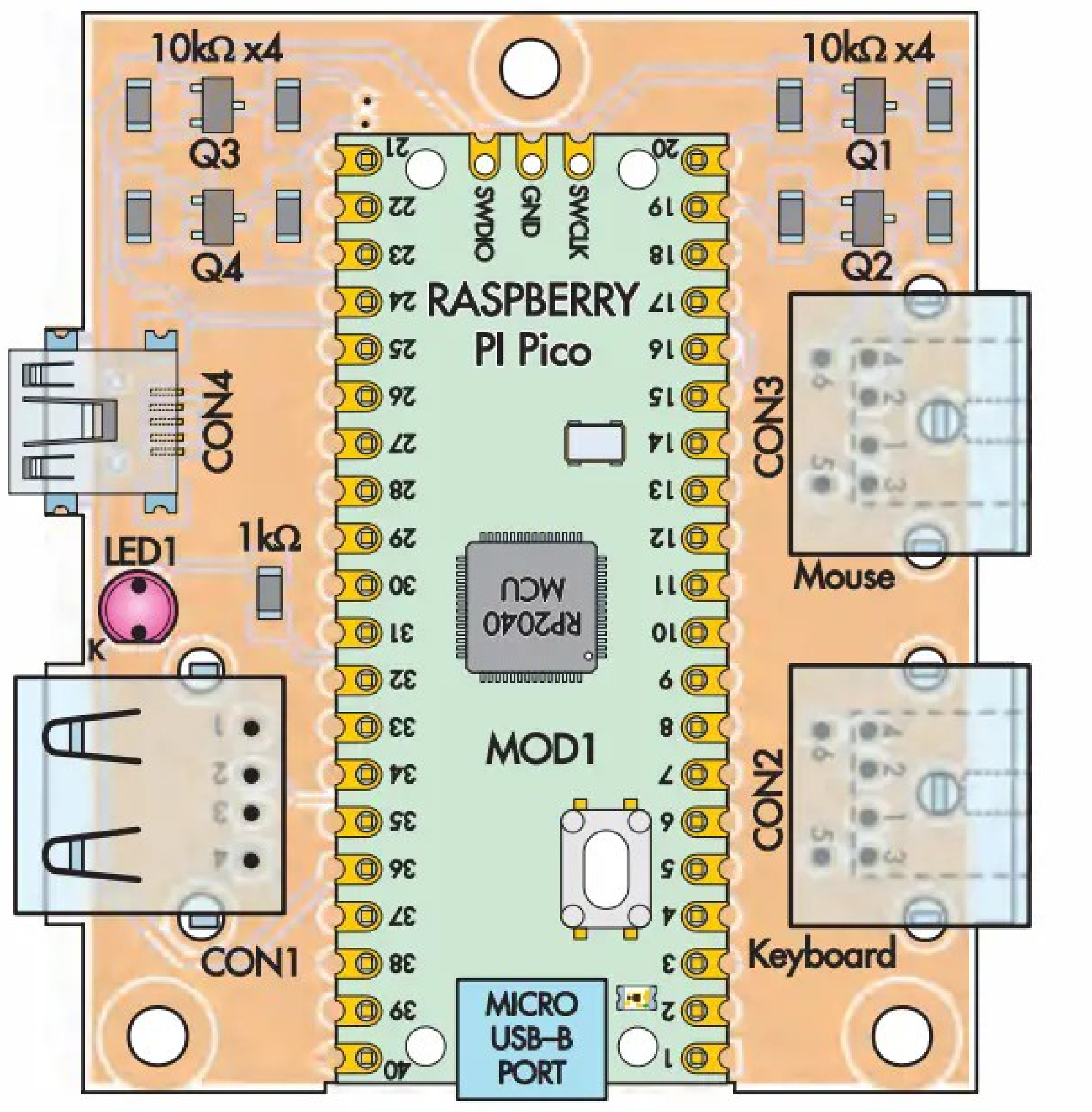


Fig.6: there are a total of 19 components to fit for the ps2x2pico version. The default firmware does not use LED1 and the 1kΩ resistor, so you can leave them off unless you plan to modify the software.

the corresponding pads on the PCB.

If you are surface-mounting the Pico, it should be possible to flow solder through the PCB and onto the pads on the Pico. You could tin the pads on the PCB and Pico with a small amount of solder to ensure that surface tension pulls the solder all the way through.

If you aren't sure, the best way to guarantee a good connection is to surface-mount some short pieces of stiff wire, such as axial lead offcuts, to the underside of the Pico. This will also need to be done if you plan to use headers to mount the Pico; it is what we did, and you can see it in the photo at upper right on page 50.

If you are using headers, solder the male headers to the underside of the Pico; the female header sockets are fitted to the top of the PCB. Connect the Pico to the PCB (by soldering the bottom of the male headers or by pressing it into the female headers).

The short wire leads should protrude through the pads in the PCB, allowing them to be soldered to it. When that is done, the CON1 USB socket and CON2/CON3 mini-DIN sockets can be fitted.

VGA PicoMite version testing

To finalise the Jiffy box version of the VGA PicoMite PS/2 Adaptor, solder the CON3 mini-DIN socket, ensuring it is down hard against the PCB. The USB socket can then only be soldered to the CON1 location.

The Adaptor should be complete enough to perform a functional test at this stage. Plug a USB keyboard (or USB wireless receiver) into the USB socket and then plug the mini-DIN cable into the mini-DIN socket.

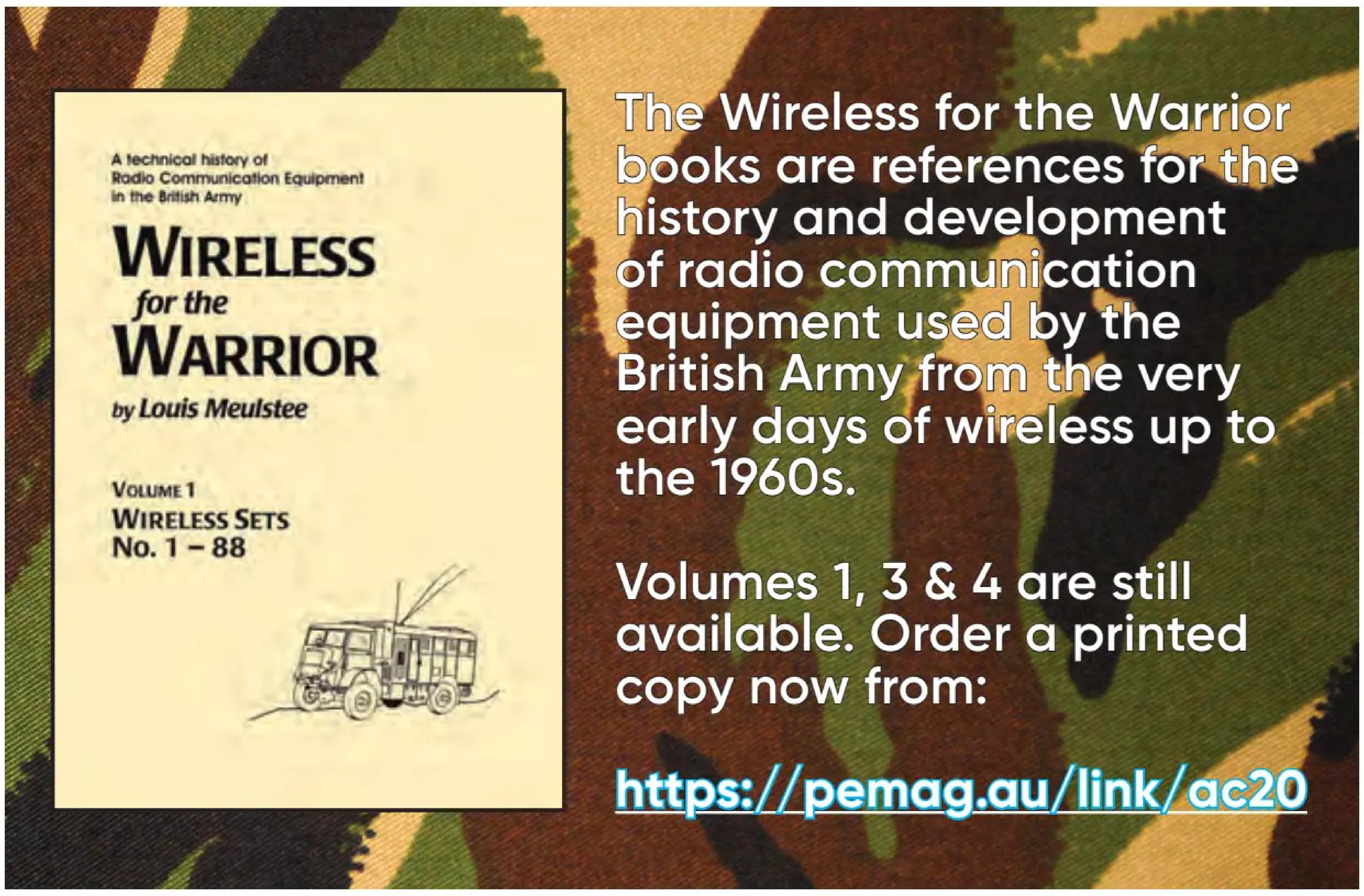
Connect the other end of the mini-DIN cable to the PS/2 socket of the computer and power it on. After a second or two, you should see the green LED light up. That indicates the Adaptor has recognised that a keyboard is attached. The LED will flicker if either the host computer or keyboard tries to communicate.

Although the LED is programmed only to light up if a keyboard is connected, we have seen some wireless mouse receivers that also cause it to happen. We suspect this type of receiver is a generic type that supports both keyboards and mouses and thus enumerates as a keyboard, even though that is not necessary for its operation with a mouse.

Jiffy box

Drill the two holes in the lid of the Jiffy Box, plus the one for the LED. Thread a machine screw from the outside and secure each with a nut inside. Each nut forms a thin spacer to keep the PCB off the lid. Then thread the PCB over the screws and fasten it in place with washers and screws.

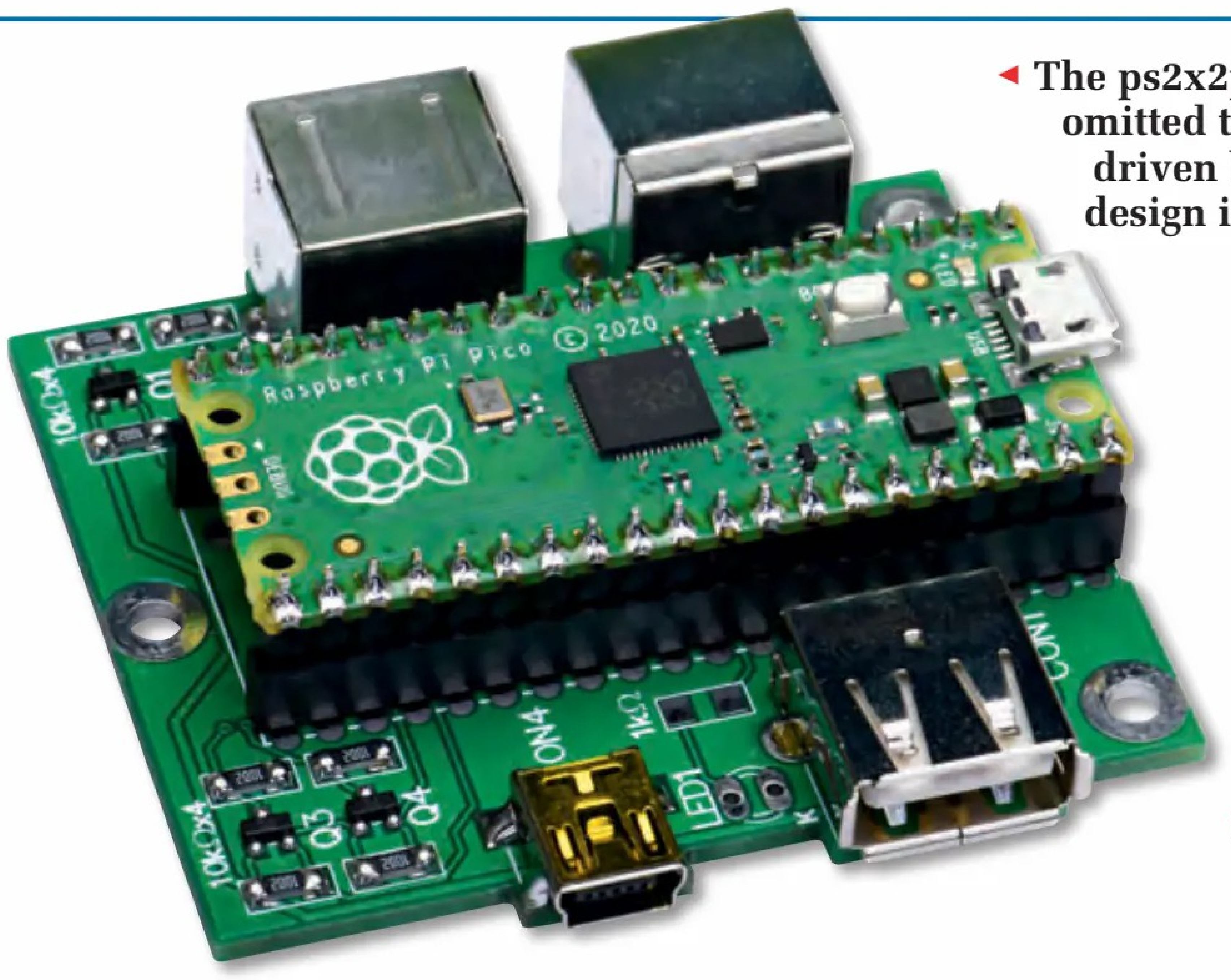
You can now make the two U-shaped cutouts in the sides of the box. Either


A technical history of
Radio Communication Equipment
in the British Army
WIRELESS
for the
WARRIOR
by Louis Meulstee
VOLUME 1
WIRELESS SETS
No. 1 – 88

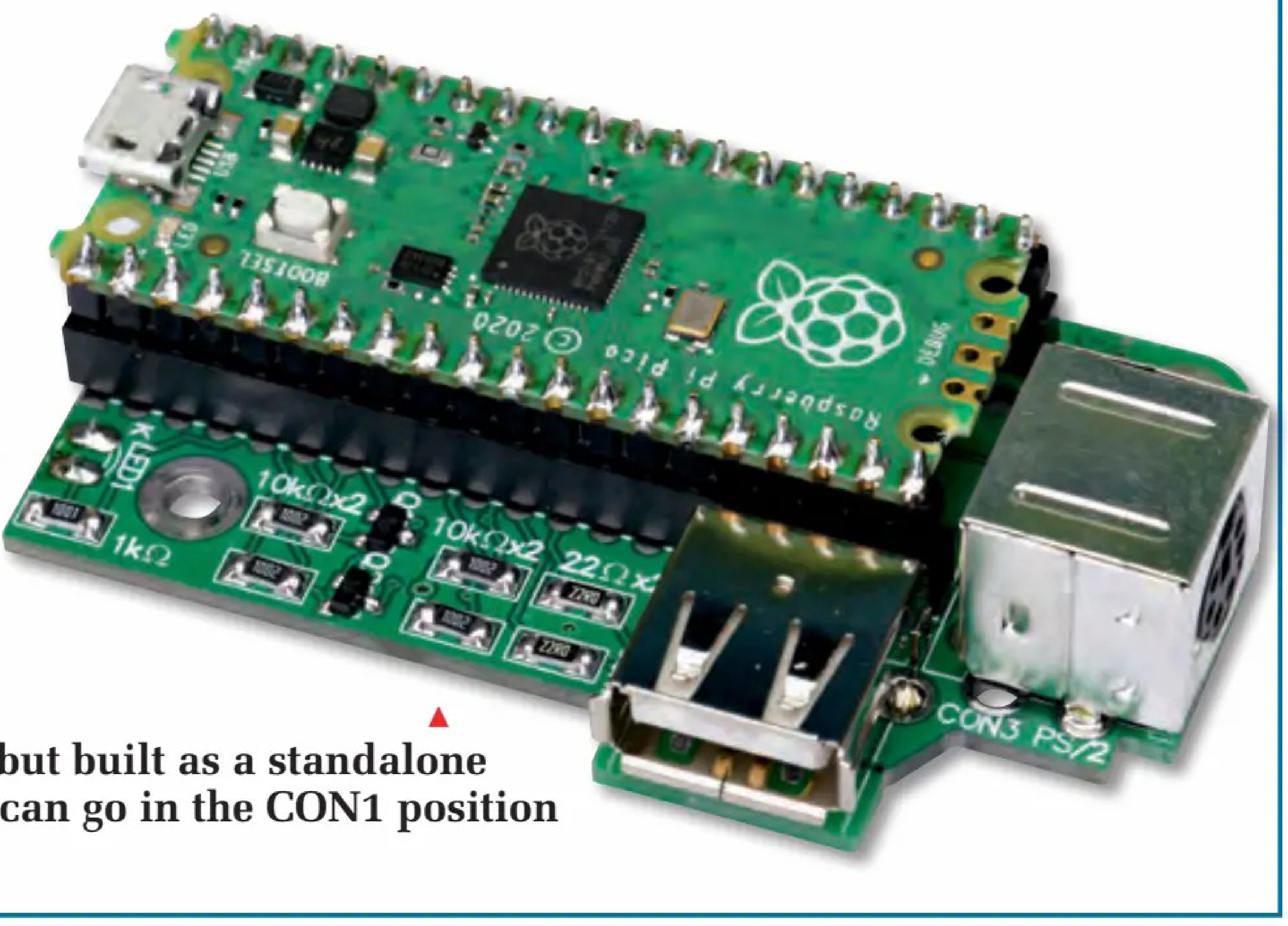
The Wireless for the Warrior books are references for the history and development of radio communication equipment used by the British Army from the very early days of wireless up to the 1960s.

Volumes 1, 3 & 4 are still available. Order a printed copy now from:

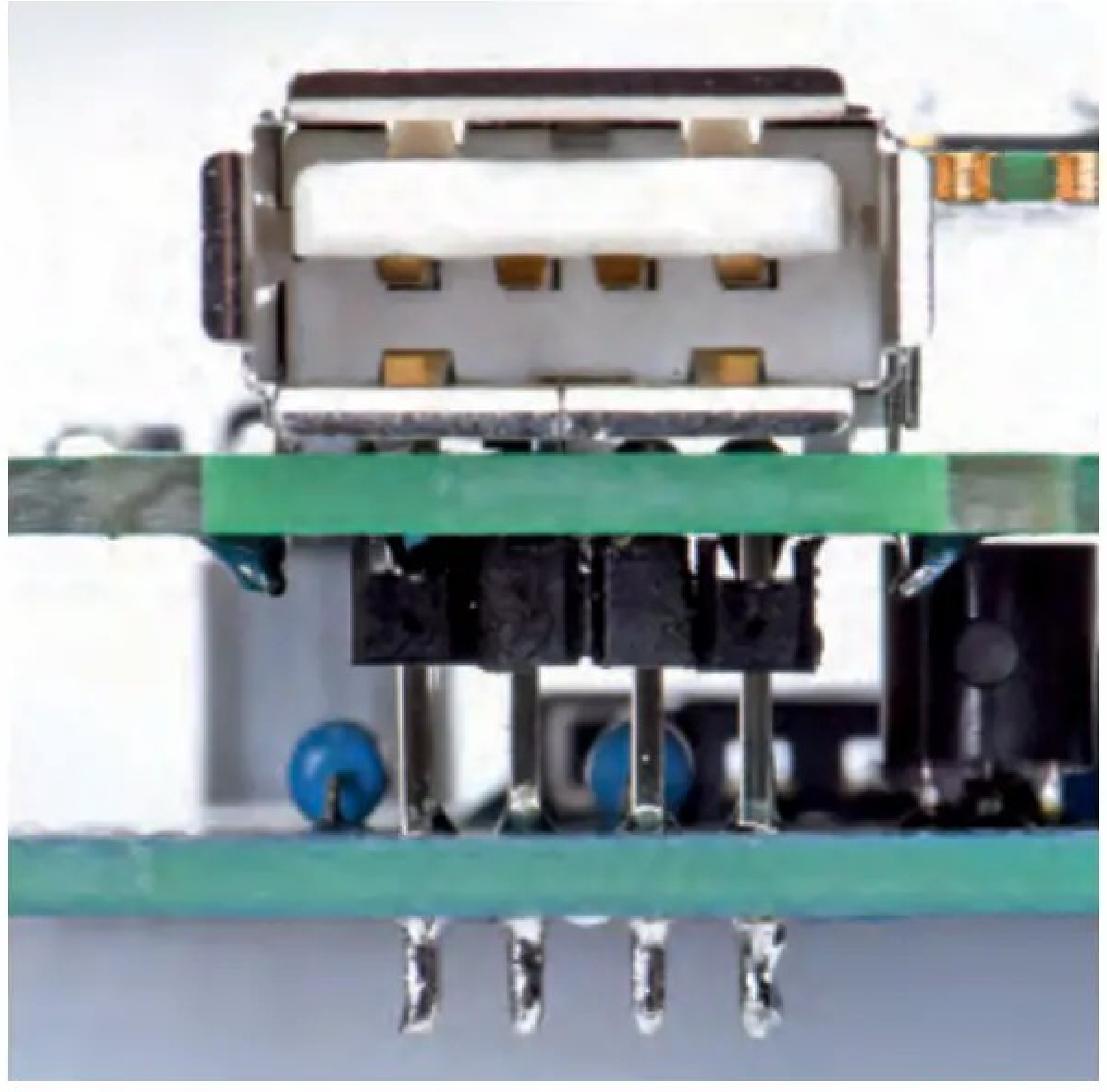
https://pemag.au/link/ac20



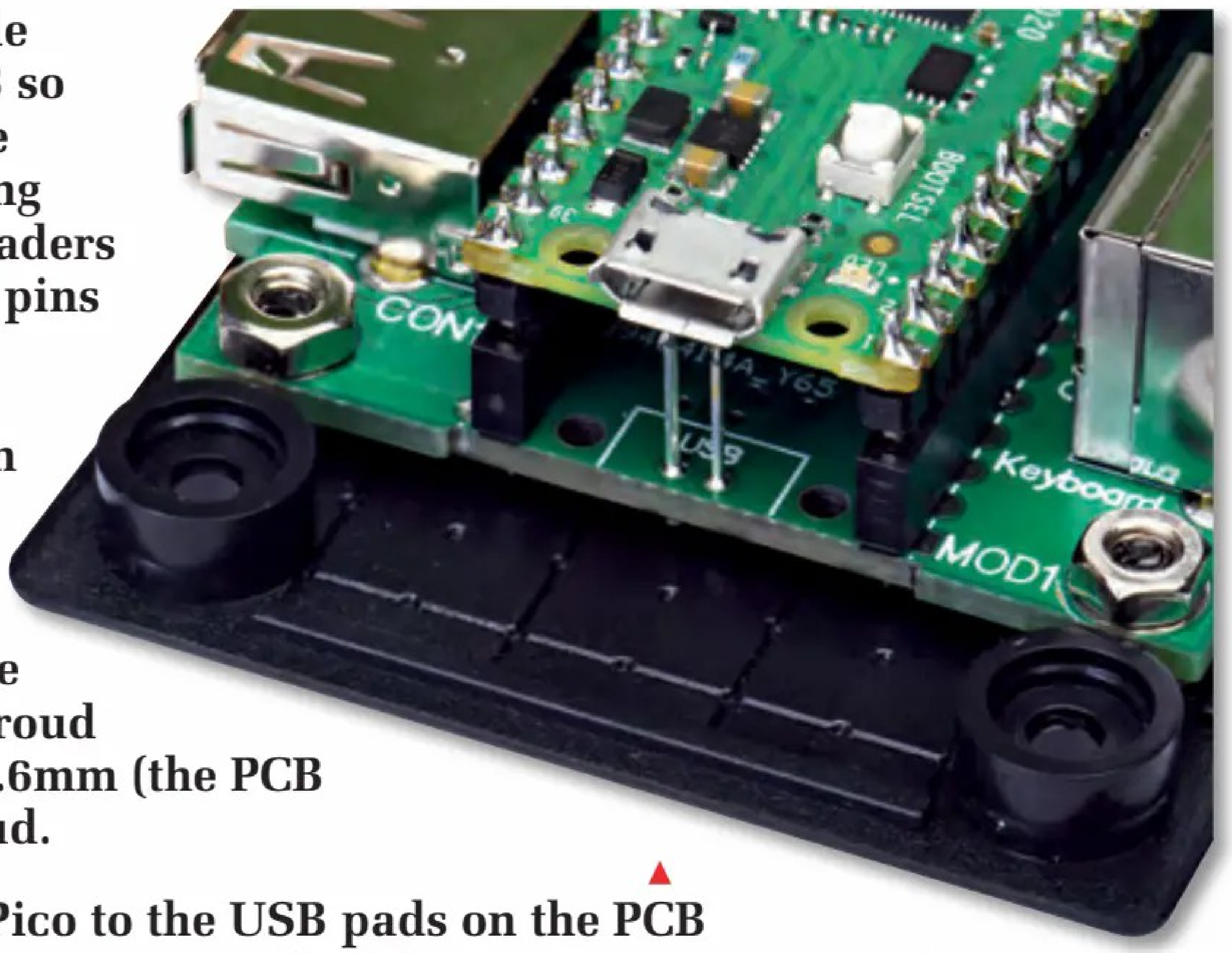
◄ The ps2x2pico version of the PCB is compact and uncomplicated. We omitted the 1kΩ resistor and the LED from our build as they are not driven by the default firmware. We decided to include them in the design in case constructors want to add support, which should not be too hard.



▲ This is the same VGA PicoMite version as shown on page 43, but built as a standalone board to be fitted into a Jiffy box. In this case, the USB socket can go in the CON1 position as the CON3 mini-DIN socket covers the CON2 position.



◄ **The headers don't protrude through the top of the PCB so that the USB socket can be soldered over the top. Using three pairs of 0.1in pin headers means that the individual pins do not come loose during soldering. The plastic has been filed down slightly on the middle pair due to the pad spacing being less than 2.54mm (0.1in) on the mini-DIN socket. We've also pushed the plastic shroud so that the pins are only 1.6mm (the PCB thickness) above the shroud.**



▲ **We used lead offcuts to directly connect the TP2 & TP3 pads on the Pico to the USB pads on the PCB underneath. Even if you are soldering the Pico directly to the PCB, we recommend soldering the two leads to the Pico.**

use our cutting diagram, or place the mounted PCB against the box to mark out the sides.

The easiest way to make these is to use a fine saw or very sharp knife (such as a hobby knife) to make the vertical cuts. Carefully score the horizontal cut on the outside of the box and then gently flex the tab with a pair of wide-nosed pliers until it snaps off.

The hobby knife can then be used to neaten and fine-tune the shape of the cutouts until the lid slots fully into place. This version is completed by screwing the lid of the Jiffy box in place.

Installing in a VGA PicoMite

Installing the Adaptor inside the VGA PicoMite is a bit more fiddly, as it requires stacking the PS/2 and USB connectors. You should have already removed the two PCB tabs by snapping them off; do so now if you have not.

Solder the pin headers to the underside of the PCB in the six pads belonging to the mini-DIN connector. Make sure that the headers do not protrude above the PCB at all.

We found the easiest way to do this was to separate the pin headers into pairs of two pins and then locate them into adjacent holes.

The plastic surrounding the middle pins may need to be filed down a bit to give clearance; unfortunately, the mini-DIN socket does not have a standard 2.54mm (0.1in) pitch between all pins.

It helps to slide the pins along the plastic so that only a small amount (about the PCB thickness) protrudes. The photo opposite shows how the PCB looks with the header pins attached.

Next, solder the USB socket above the PCB in the location marked CON2 and attach the 6mm spacer next to the LED. The Adaptor PCB can now be slotted onto the VGA PicoMite. Make sure that it is square and does not contact any components underneath.

Solder the six pins to join it to the VGA PicoMite's PCB. You might find that you need to slightly enlarge the back panel hole for the USB socket, since it is larger than the mini-DIN socket.

You will also need to drill a hole for the LED in the front panel. Fig.7 shows the suggested panel modifications for those two holes.

Connect a USB keyboard, then power on the VGA PicoMite. The LED should light up, then flicker as the PicoMite initialises.

ps2x2pico version testing

The Adaptor can be tested by connecting it to a PS/2-compatible computer using a pair of 6-pin mini-DIN cables. Connect the keyboard and mouse to the USB socket using a hub, if necessary.

We imagine many people will use a wireless keyboard and mouse combination, in which case a single compact receiver is all that needs to be plugged into the USB socket.

The mini-USB socket is provided in case extra power is needed; the socket on the Pico cannot be used to supply power as it is working in host mode.

Attach the machine screws to the lid of the Jiffy box and secure them on the inside using three of the nuts. Slide the PCB over the screws, then secure it to the lid with washers and screws.

Use the Fig.4 cutting diagram to make the U-shaped slots in the side of the Jiffy box to accommodate the

Parts List – ps2x2pico USB to PS/2 Converter

1 double-sided PCB coded 07111232, 58 × 52mm
1 Raspberry Pi Pico programmed with ps2x2pico.UF2 (MOD1)
2 20-way pin headers (**optional**, for MOD1)
2 20-way female header sockets (**optional**, for MOD1)
1 PCB mount USB-A horizontal socket (CON1)
2 6-pin mini-DIN sockets (CON2, CON3)
1 surface mounting mini-USB socket (CON4)
1 UB5 Jiffy box
3 M3 × 10mm panhead machine screws
6 M3 hex nuts
3 M3 flat washers
2 short pieces of wire (if mounting the Pico on headers)
2 6-pin mini-DIN to 6-pin mini-DIN cables

Semiconductors
1 3mm green LED (LED1; **optional**)
4 2N7002 N-channel Mosfets, SOT-23 (Q1-Q4)

Resistors *(all M3216/1206 SMD, ¼W)*
8 10kΩ 1 1kΩ (only needed it fitting LED)

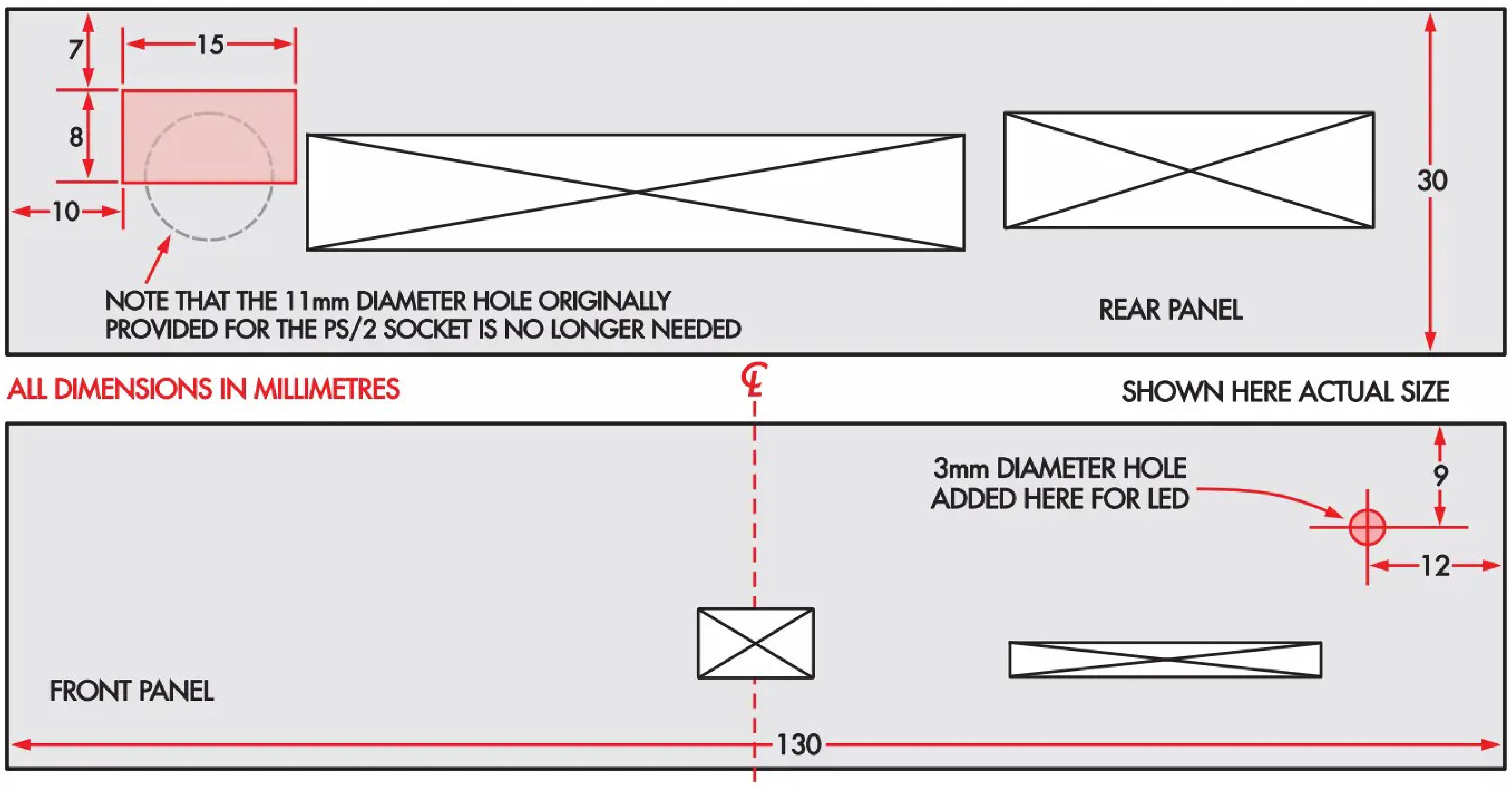

7
15
8
10
NOTE THAT THE 11mm DIAMETER HOLE ORIGINALLY PROVIDED FOR THE PS/2 SOCKET IS NO LONGER NEEDED
REAR PANEL
30
ALL DIMENSIONS IN MILLIMETRES
℄
SHOWN HERE ACTUAL SIZE
3mm DIAMETER HOLE ADDED HERE FOR LED
9
12
FRONT PANEL
130

Fig.7: this diagram shows modifications to the H0376 instrument case used for the VGA PicoMite. The added LED for the PS/2 Adaptor sits on the front panel near the SD card socket, while the cutout for the USB socket overlaps the original location of the mini-DIN socket on the rear panel.

connectors. A sharp hobby knife or fine-toothed saw are good choices here. Finally, attach the lid to the box with its included screws.

Jiffy box labels

Figs.8 & 9 show panel artwork that can be applied to the lids of the Jiffy boxes. Fig.9 helpfully marks the distinction between the keyboard and mouse sockets.

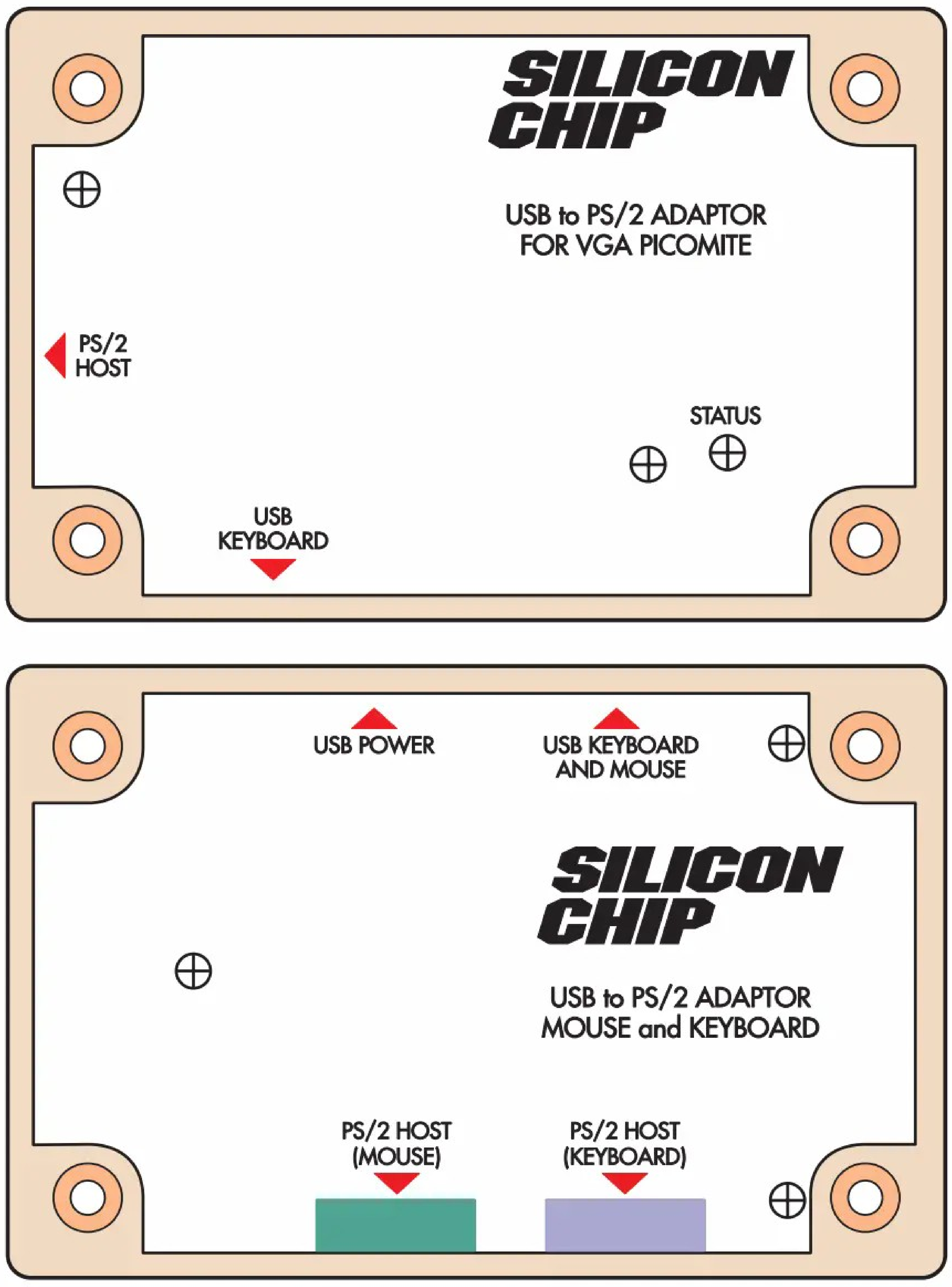
Conclusion

We continue to be impressed by the capabilities of the Raspberry Pi Pico, and this application is a perfect use for its abilities.

We're sure there are many readers out there with PS/2 equipment who will make use of the option to use a modern USB keyboard on their legacy devices.

You can find a list of helpful links listed below:

- github.com/No0ne/ps2x2pico/
- wiki.osdev.org/PS/2_Keyboard
- wiki.osdev.org/PS/2_Mouse **PE**


SILICON CHIP
USB to PS/2 ADAPTOR FOR VGA PICOMITE
PS/2 HOST
STATUS
USB KEYBOARD
USB POWER
USB KEYBOARD AND MOUSE
SILICON CHIP
USB to PS/2 ADAPTOR MOUSE and KEYBOARD
PS/2 HOST (MOUSE)
PS/2 HOST (KEYBOARD)

Figs.8 & 9: this simple label (shown at top) can be affixed to the top of the Jiffy box for the VGA PicoMite Adaptor. The label below will help users differentiate between the otherwise identical mouse and keyboard sockets. We've omitted the hole for the LED as we expect most readers will not use it. Both labels are shown at actual size.

ENIGMA
COMPONENT SHOP
www.enigma-shop.com
Electronic Components for the Hobbyist!
Cable & Wire • Capacitors • Connectors • Filters
Kits & Modules • Hardware & Cooling
and much more...
READY TO BE REWARDED?
Join our loyalty scheme and **earn points** on every purchase you make which can be **used for discounts** in the future.
Read our reviews on Trustpilot

Raspberry Pi Clock Radio

As described last month, this new Clock Radio can also act as a media player and has many alarm options to ensure you wake up at the right time and in the manner you prefer. You can even synchronise the alarms on multiple Clock Radios. Since we've already explained how it works, this article will cover construction and how to use it.

Part 2 by Stefan Keller-Tuberg

The major functions of the Clock Radio are handled by a Raspberry Pi (model 2, 3, 4, Zero W or Zero 2W). However, we need to add some extra hardware for the time LED display, buttons, switches, audio amplifier to drive the speakers and so on. This additional hardware is hosted on two custom PCBs that mount at right angles to each other, with a series of soldered fillets between them.

The construction process will therefore be to mount the components on these two boards, join them together, wire them up to the Raspberry Pi (via a ribbon cable with IDC connectors) and then mount that assembly in the case. Once it's in the case, it can be wired to all the external switches, connectors etc.

The software will function regardless of which bits of additional hardware are connected, so any parts you don't need can be left off the PCB without needing to modify the software. For example, if you don't want the seconds digits, you don't need to fit that two-digit LED display.

Construction

We suggest you start by fitting the SMD components to the Display board. This is built on a 150 × 44mm PCB coded 19101242; its overlay diagram is shown in Fig.4.

Fit the 48 current-limiting resistors; if you are unsure how well the brightness will match between the larger and smaller displays, you can start by fitting just the 32 resistors associated with the 0.8-inch displays. Either way, it's easier to solder them in groups.

If you're using different 3mm LEDs than the two options in the parts list, leave the two 1.3kΩ colon LED series resistors out for now; otherwise, add them at this stage.

It's easier to solder surface mount components by applying flux to the PCB before positioning the component. When all SMD resistors are soldered, you can clean off the residual flux with isopropyl alcohol, methylated spirits or a specific flux residue cleaner. Stubborn areas can be cleaned with a toothbrush or lint-free cloth.

Do this before installing the other components, as you don't want flux residue to be transferred onto the optical surfaces. Cleaning the board once through-hole parts have been fitted is also much harder.

Move on to the 7-segment displays. Make sure you have a clean work surface, as it's easiest to do this with the front of the displays resting on it. Mount the two 0.8-inch displays on the opposite side of the PCB than the resistors, with their decimal points towards the edge connections.

Solder two opposite corner pins first, so you can adjust the displays to be tight against the PCB and nicely horizontal. When you are satisfied with that, solder the remaining two corner pins before completing the others.

The 0.56-inch display has a lower profile than the others, so mount it proud of the PCB by a few millimetres to keep its front face in the same plane as the others.

Insert the smaller display into its holes, check that it is correctly oriented with the decimal points towards the edge connector and then turn the whole assembly face down onto a flat surface. Press down on the assembly to ensure that the two larger displays are hard against the bench, and you will see that the legs of the 0.56-inch display barely protrude through the PCB.

Solder two of its opposite corners as before, then inspect. You can use a straight edge, such as a ruler, to ensure that the front faces of all three displays are aligned in the same plane, adjusting the small one if required until you are satisfied. Then solder the remaining pins as before.

Insert the 3mm LEDs into the gap between the larger displays. The cathodes (the shorter leg) are towards the edge connector. Again, place the assembly face-down onto a flat surface and solder one leg only on each 3mm LED. Adjust their angles and heights until you are happy before soldering the other leads.

Raspberry Pi Clock Radio, part two

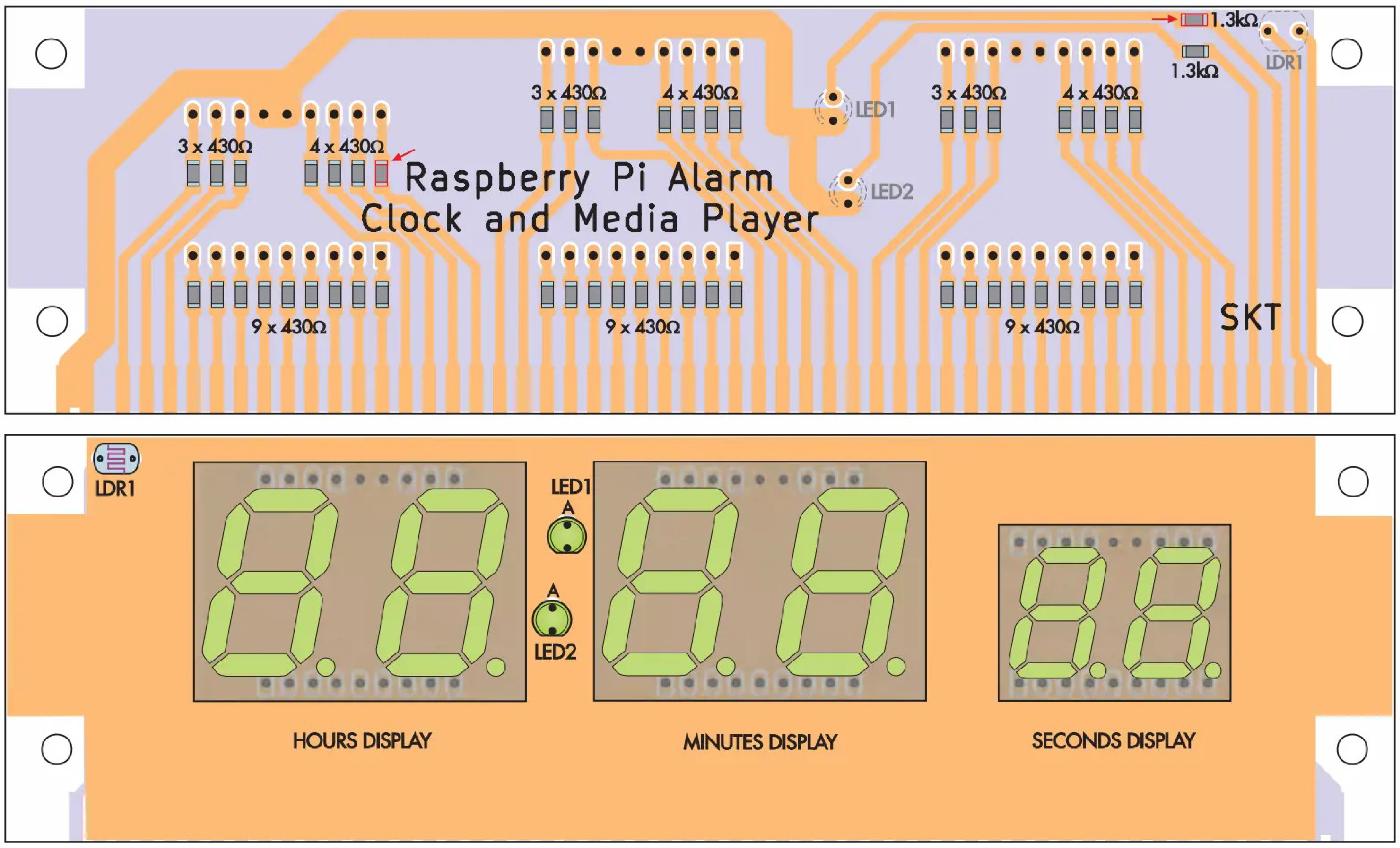


Fig.4: the display board has three 7-segment displays, two colon LEDs and an LDR on the front, while the current-limiting SMD resistors are on the back. It's easiest to fit the resistors first, but see the comments in the text about possibly needing to change some of their values depending on which LED displays you use.

The two resistors marked with a red arrow can be changed to test the colon LED and the brightness of the small 7-segment display.

Fit the LDR next. The procedure is similar to the 3mm LEDs, but the LDR orientation doesn't matter.

Main board assembly

The main board is coded 19101241 and measures 150 × 83mm. Fig.5 is its overlay diagram.

Before fitting the components, it is worthwhile to verify that the supply rails have not been inadvertently shorted during the PCB fabrication process. Using a multimeter, check for an open circuit between pins 10 and 20 of the 74HCT374 (the diagonally opposite pins) and between pins 8 and 16 of the 74HC139 next to it (also diagonally opposite).

It is rare, but if either is shorted, you just saved yourself a headache trying to find an almost impossible-to-find problem after the board is assembled!

Start with the surface-mounting amplifier IC (IC13), carefully installing it with the correct orientation. Pin 1 on the chip is marked with a dot or divot and on the PCB silkscreen. Apply flux to the PCB and then position the chip.

Solder just one corner pin first and then check that all of the pins align with their pads and the chip is more-or-less centred over the pad area. If not, soften the solder and nudge the chip until it's properly aligned. When you're happy, dab solder on the opposite corner and recheck the orientation. Solder the remaining two corner pins before completing the rest.

Clean up any bridges with solder wick, then remove any residual flux as you did with the display board.

Continue by fitting the through-hole components, beginning with the lowest profile devices. Install the three diodes (watching their cathode stripe polarity) and the resistors, then the IC sockets (notched ends towards the bottom or right), followed by the ceramic or MKT capacitors, except the 100nF capacitor near the top edge of the board.

Mount the electrolytic capacitors next, with their longer (positive) lead towards the + symbols. The striped side of the can is negative. Solder the relay next, followed by the transistors from shortest to tallest, except Q2, ensuring they face as shown in Fig.5.

Don't install chips into their sockets just yet, we will do that once the two boards have been joined. You can fit CON5 and the 40-pin header that will go to the Raspberry Pi now, but leave off the polarised headers around the board's edge. That will give you reasonable access so you can join the two boards.

Joining the boards

If you've never assembled two boards using solder joints at right angles, a little planning will simplify the task. Ideally, the two boards should be aligned close to a right angle, but an error of a degree or two either way will not make much difference.

I found that the easiest way to join the boards was to raise the main board above the bench using four PCB standoffs. Using a plastic right angle or similar, carefully tape the main and display boards individually to the right angle, as shown in the photo shown overleaf.

Overvoltage protection can trip with poorly regulated supplies

One of the 5V plugpacks I pulled from my box of spares measured 5.45V with no load. I figured this plugpack would be OK because the Clock's protection circuit kicks in at around 5.65V – well above the supply's no-load output.

However, when I actually connected the plugpack, nine times out of ten, I could see the Pi LEDs flash briefly before turning off, a sign that the clock's protection circuit had kicked in. An oscilloscope revealed that this dodgy plugpack was not very well regulated; at the moment a load was applied, the plugpack's voltage would fluctuate enough to trigger the protection.

I found a simple workaround with a spare 10,000μF electrolytic screwed into the terminals of the clock's power connector, which smoothed that dodgy plugpack's ripple enough not to reach the protection-tripping threshold. I have not had this happen with any of my other 5V plugpacks, so we recommend not using a dodgy plugpack in the first place.

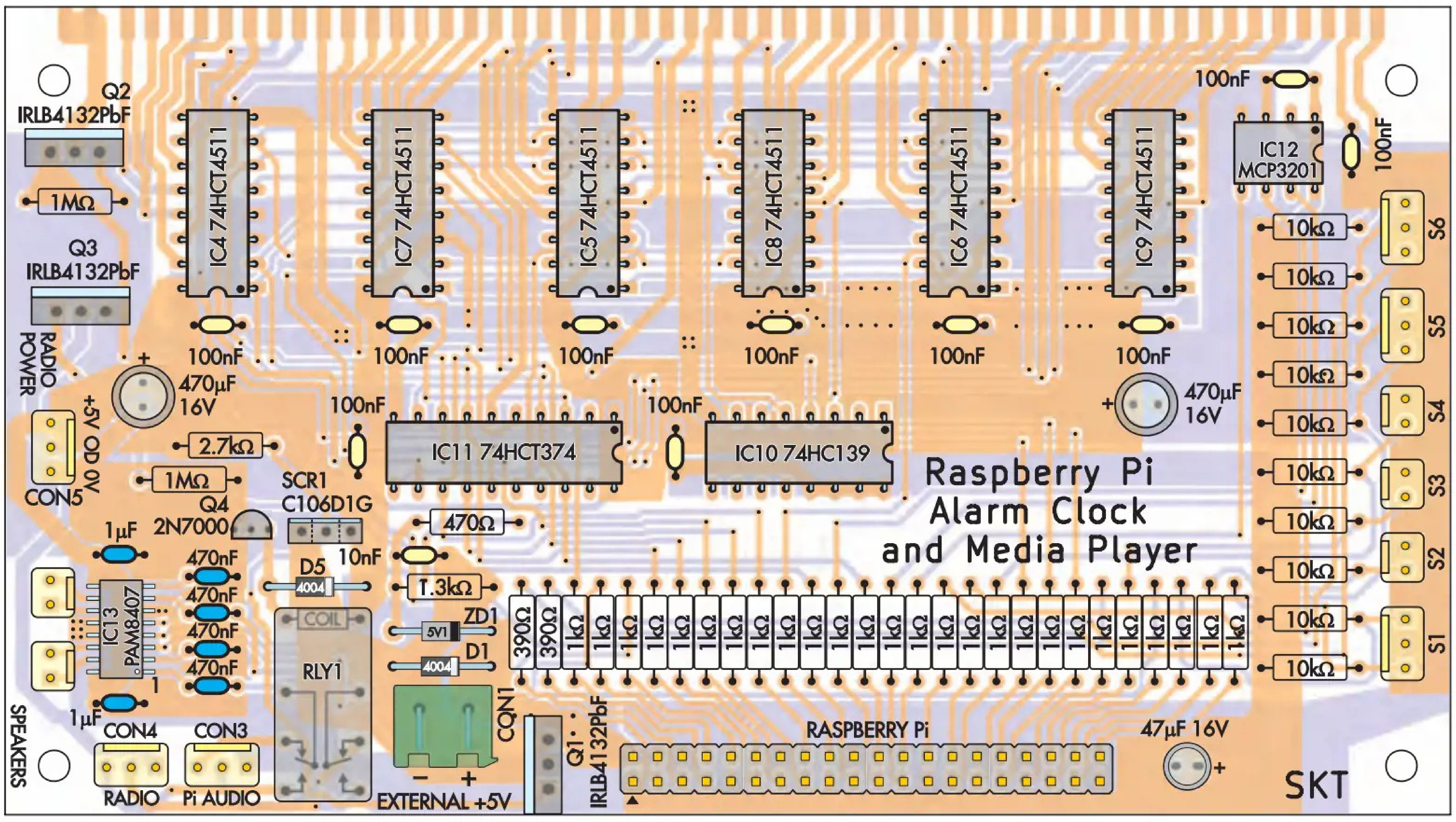

Q2
IRLB4132PbF
1MΩ
Q3
IRLB4132PbF
RADIO POWER
+5V OD 0V
CON5
IC4 74HCT4511
IC7 74HCT4511
IC5 74HCT4511
IC8 74HCT4511
IC6 74HCT4511
IC9 74HCT4511
IC12 MCP3201
100nF
470µF 16V
2.7kΩ
1MΩ
SCR1 C106D1G
Q4 2N7000
1µF
IC11 74HCT374
IC10 74HC139
470Ω
10nF
1.3kΩ
ZD1 5V1
D5 4004
D1 4004
COIL
RLY1
470nF
IC13 PAM8407
SPEAKERS
CON4
CON3
RADIO
Pi AUDIO
EXTERNAL +5V
CON1
Q1 IRLB4132PbF
RASPBERRY Pi
390Ω
1kΩ
10kΩ
47µF 16V
S1
S2
S3
S4
S5
S6
SKT
Raspberry Pi
Alarm Clock
and Media Player

Fig.5: the components on the main board are nearly all through-hole parts. The solder pads along the top of this board will connect to those along the bottom of the display board later. When assembling this PCB, watch the orientations of the ICs, transistors (including Mosfets and the SCR), diodes and electrolytic capacitors.

Ensure that the display board evenly overhangs the solder side edge of the main PCB by about one millimetre for the entire length of the mating edge. The slight overhang is so you can add solder beads to the pads on either side of the main board, as there are electrical connections on both sides. That also provides extra mechanical strength.

Solder the first joint on the left end of the mating edge, then the one at the right end. For these first two joints, only solder the top side of the main board. This will be fragile, but it allows you to check and easily adjust the alignment of the display board with the main board.

If the alignment is imperfect, soften one joint at a time, alternating from one to the other, and make a sequence of minor adjustments until you're happy.

When satisfied, add another three or four solder joints spaced out along the top side to secure it, then flip the assembly over and work on the other side. Solder the opposite ends to shore up the mechanical strength, then solder all the remaining connections. Finally, flip it back over and finalise the original side. The boards will now be very firmly bonded, and you can remove the right angle.

Complete the electronics by adding Mosfet Q2, the 100nF capacitor in the top-right corner of the main board and all the polarised headers.

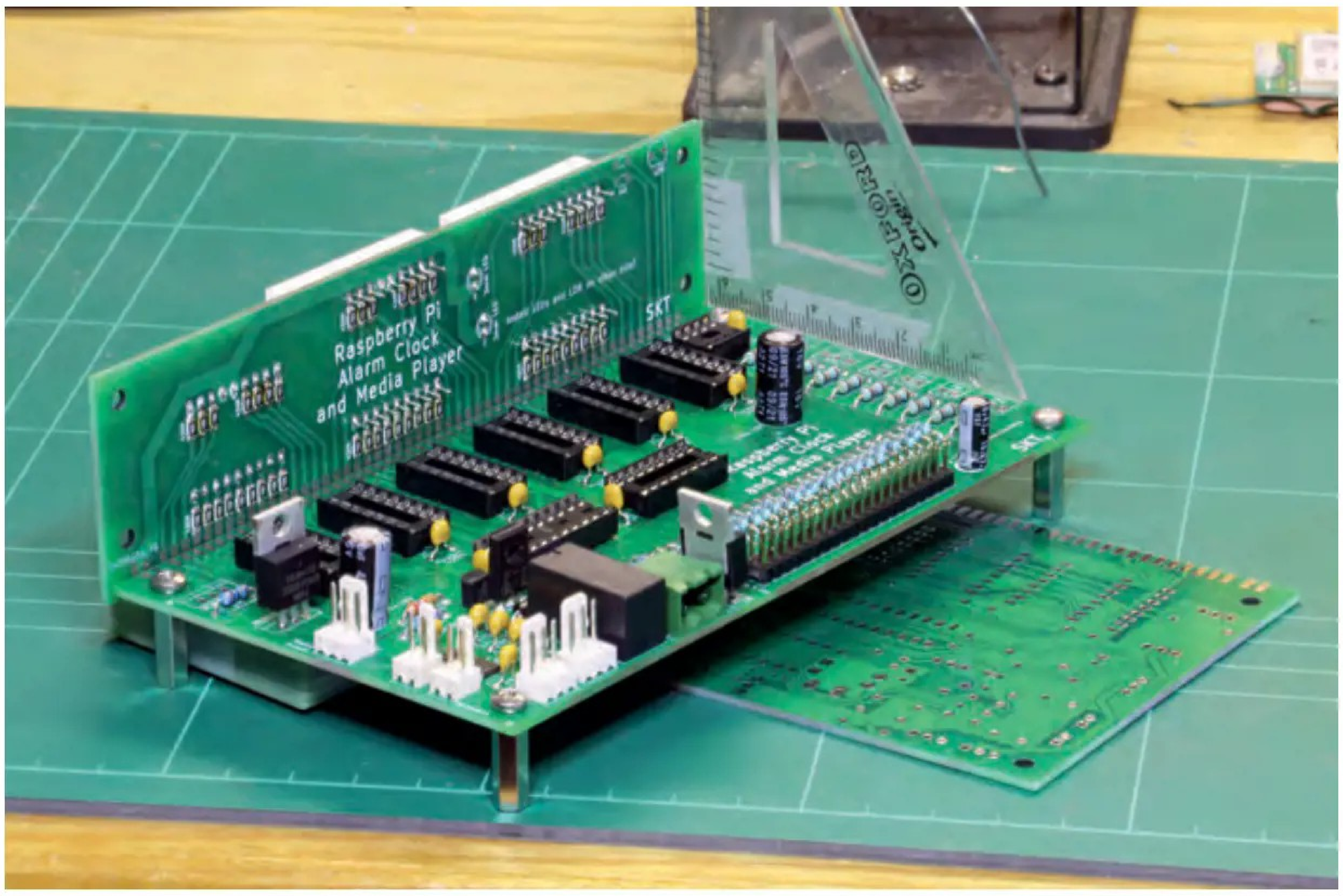
Final assembly

The 40-conductor ribbon cable for connecting the Raspberry Pi to the clock board can be made from scratch, or you can use an old IDE hard drive cable if you have one. To make it from scratch, carefully position an IDC crimp connector within a vice's jaws and protect it using small timber offcuts.

When you slide the 40-conductor cable into the connector, be careful to align it at right angles. It is easy to accidentally crimp the connector at an angle, in which case the cable probably won't work. As you tighten the vice, you'll hear the two ends of the crimp connector click into place, at which point you should ease off the pressure.

Old IDE cables can be used for this application, but if the old cable supports two IDE drives, cut off one end of the cable using a Stanley knife. Before you do, double-check that the two crimp connectors have all 40 holes open; some IDE cables have one blocked hole, and those won't be any good for this project.

Next, you need a cable with a 3.5mm stereo plug on one end and a 3-pin polarised header plug on the other. I cut the end off an old 3.5mm headphone cable, stripped back the insulation and crimped the wires to the header pins. I then applied some flux paste, dab-soldered the signal wires and shield, then inserted the pins into their housing.



I found that raising the display board was helpful when joining it to the main PCB. The boards can then be taped together and soldered on both sides.

Raspberry Pi Clock Radio, part two

Because headphone conductors are fragile, I used a sparing amount of hot glue to neatly encapsulate the top of the housing, the loose conductors and the end of the headphone cable. This provides some strain relief; you could also use neutral-cure silicone sealant.

Double-check the orientation of the transistors, diodes and electrolytic capacitors on the main board. When happy, carefully install the socketed ICs, noting their orientations & types, and ensuring all pins have correctly entered the sockets.

The three chassis-mounting momentary pushbuttons (S2-S4) connect to the main PCB via two-way cables terminated in polarised header plugs, so solder the wires to them and crimp/solder the plugs on the other end. Similarly, prepare three centre-off momentary toggle switches with three-way cables (that could be stripped from ribbon cable) terminated in three-way polarised header plugs.

The banana sockets/binding posts for the speakers (two positive and two negative) are wired to another two polarised header plugs, this time with slightly heavier-duty wire. However, it can't be too thick, or you won't be able to get it into the plugs. Medium-duty hookup wire should work. Wire one red and one black terminal to each plug. The polarity doesn't matter as long as it's the same for both.

For the external 5V supply, solder wires to the chassis socket and then screw them into the pluggable terminal. The positive input should be on the right when plugged into the PCB.

If adding a radio receiver input, its audio output goes to a three-way polarised header plug that follows the same arrangement as for the Pi audio input. The GND connection is closest to the relay and the left and right channels sequentially on the other two pins. If your audio source is mono, short the left and right inputs and feed them from the mono source. Connect its power supply to CON5 as per one of the options in Fig.3 last month.

Testing the Clock

We recommend installing the clock software onto the Raspberry Pi (if you haven't already) and ensuring the software works before testing the hardware, as explained last month.

Commence hardware testing by repeating the earlier multimeter test to verify that the power supply rails are not shorted. The rails will not be completely open-circuit now because there are capacitors and ICs on board, but the rails should not be dead shorted either.

You can test the overvoltage protection circuit if you have a current-limited bench supply. If you don't, we recommend you don't test it and assume it works because if you apply a high voltage and it doesn't work, you will fry everything.

If you have the bench supply, the test procedure is as follows. Use a voltmeter to accurately measure the applied voltage directly at the variable power supply (if it has an onboard meter, you can use that). Tenths of a volt make a difference, so it is important that the voltmeter is accurate and measures to at least 100mV resolution!

Use another voltmeter to monitor the voltage between the 0V power input on the board and the tab of Mosfet Q1.

Starting low, ramp up the variable power supply to around 4.5V, then slow down. As the input approaches 5V, minimal voltage should be registered on the second voltmeter at the tab of Q1, indicating that the Mosfet is switched on, its default during regular operation. Continue slowly increasing the variable power supply so that its output rises just above 5V.

The protection threshold voltage will depend upon the ambient temperature and idiosyncratic characteristics of the 5.1V zener diode and the SCR. When the variable input voltage reaches 5.6-5.8V (certainly below 6.0V), a voltage should develop on the Mosfet's tab, indicating that Q1 has switched off.

Don't wind the variable power supply past 5.85V! If the SCR has not fired by 5.85V, switch off and look for a problem in the protection components above and to the right of RLY1.

When satisfied that Q1 is switching off as expected around 5.6-5.8V and before powering down the variable supply, verify that the voltage across the 2.7kΩ resistor is about the same as the variable supply output, confirming that the SCR has fired.

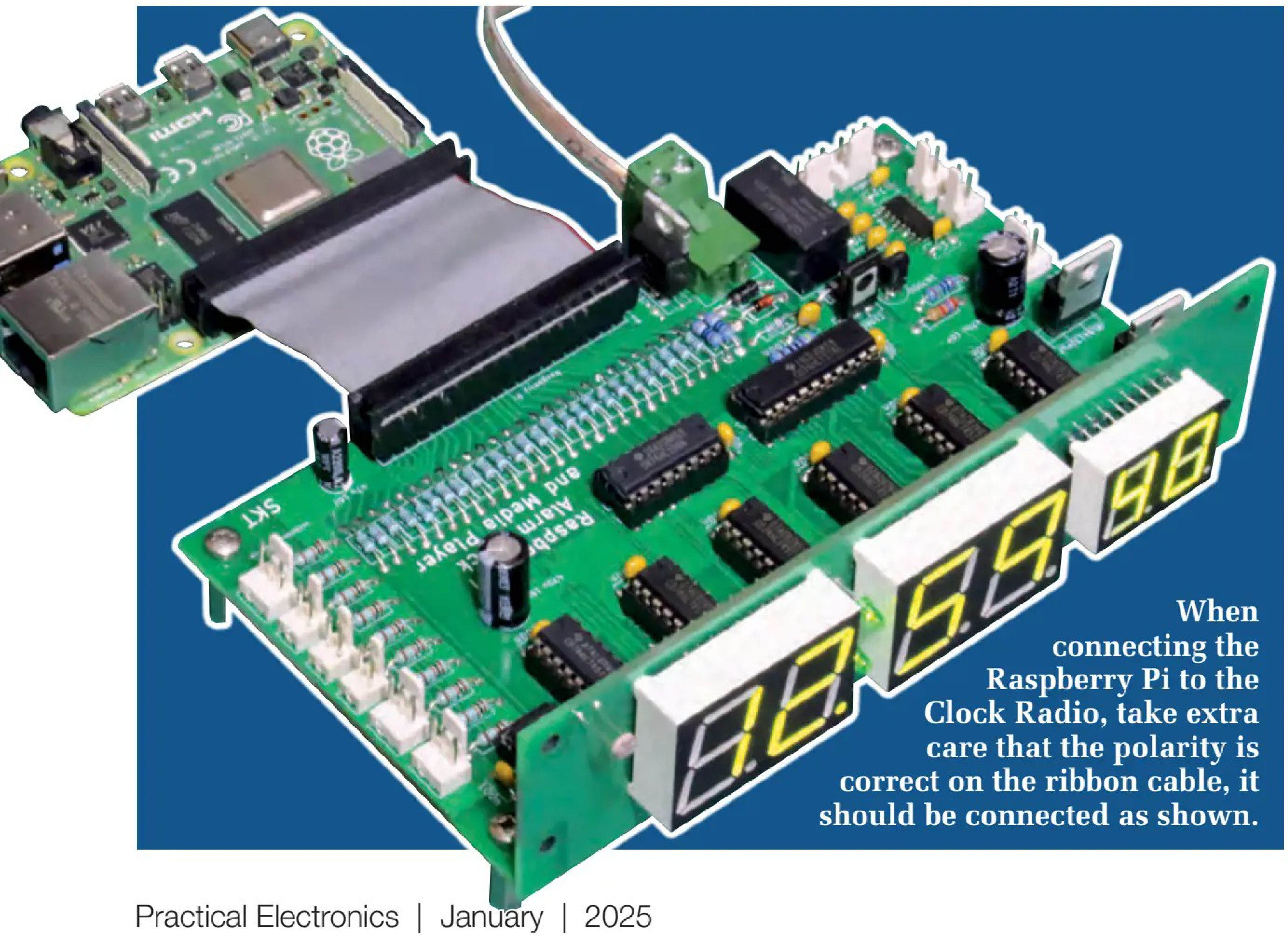
Finishing the wiring

Now you can plug everything into the main board. That includes the buttons, switches, the audio cable from the Pi's output jack socket to the three-pin polarised header on our board, the external 5V supply, the speaker terminals and the optional radio board.

Also connect the Raspberry Pi via the ribbon cable, being very careful to get its polarity right. Refer to the adjacent photo and Fig.6 to see how it's done. Pin 1 of the Pi must be wired to the bottom-left pin of the header on the main board; use a DMM set on continuity mode to verify that before applying power.

When you apply power to the assembly now, do it via the clock board only and don't connect a separate power supply to the Pi. The Pi will receive its power via the clock assembly through the ribbon cable.

When power is first applied to the



When connecting the Raspberry Pi to the Clock Radio, take extra care that the polarity is correct on the ribbon cable, it should be connected as shown.

whole assembly, the display will start blank, and the Pi will go through its booting process. You should see the Pi's power LED turn on as the Pi boots, but the clock display will remain blank for around half a minute. When the display eventually illuminates, it will show an incorrect time.

The Raspberry Pi will be trying to connect to your home network, and when it does, it will grab the current time from the internet. The display will then jump to the correct time and will be time-locked from that moment onward.

If the 7-segment displays remain blank, double-check the LED brightness and threshold settings on the web setup page. It's possible that they have been dimmed very low and just need to be brightened. Otherwise, use ssh (eg, via PuTTY) to connect to the Pi and use the alarm clock debugging modes described in the software installation documentation to check the software's health.

The Pi installation script should have been run to completion without errors. If not, that might be a clue as to why it isn't working.

If the display is still not illuminated, but the software appears to be running, use a logic probe, oscilloscope or frequency counter to check that there is activity on the data and address bus GPIO lines between the two boards and on the PWM GPIO line to the gate of Mosfet Q2. Also check that you have used the correct Mosfet types (they must handle logic-level drive).

Matching display LED brightnesses

If you have not yet installed current-limiting resistors for the two colon (":") LEDs or the small display, you can start to experiment. Turn off the clock and tack-solder candidate resistors for just one each. Segments within an individual display will have the same brightness, so once you've matched one to the rest, that resistor value should work throughout.

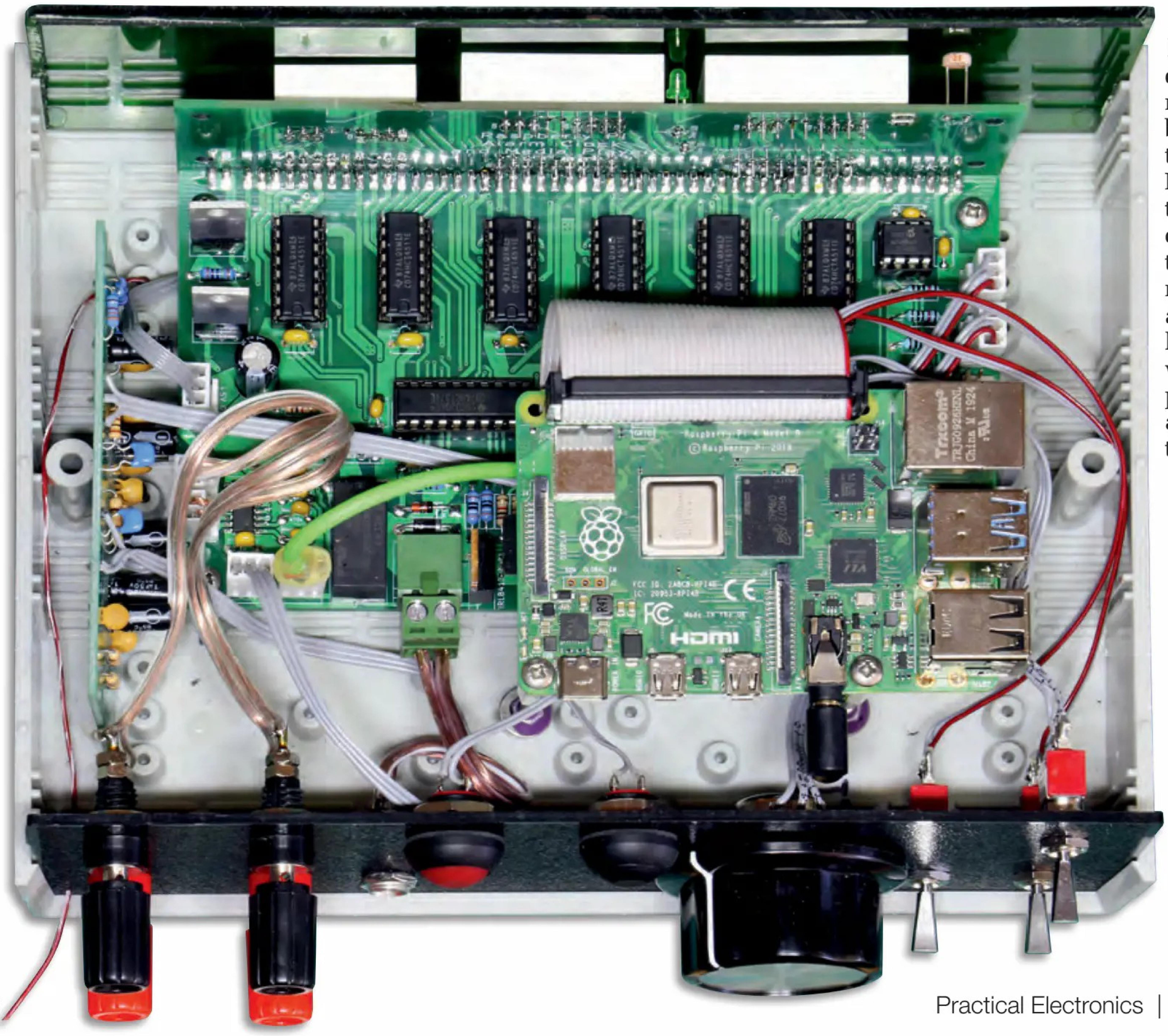
It is easier to use axial leaded resistors rather than SMDs for these experiments unless you have a sample book of suitable SMD resistors. Start with the default values of 430Ω for the display and 1.3kΩ for the colon or 470Ω/1.2kΩ if you don't have those other values.

A convenient resistor to choose for the small display is the top one closest to the larger digits because it illuminates a segment alongside the larger displays, making it easier to compare. Both colon resistors are convenient to access. Refer to the two resistors marked with a red arrow in Fig.4.

Power on the Pi, go to the web setup page and put the clock hardware into a darkened room. Adjust the minimum brightness slider down to a level where the illuminated LED segments are just visible, then set the maximum brightness slider to midrange. Remember to save the setting using the web save button, or the next step won't work as expected.

Choose the "Start the LED brightness test" web button to turn every LED on (displaying all 8s) and force the display to the minimum brightness setting for a moment. You'll have to judge whether the test segment and the test colon LED are the same, brighter or dimmer than the large numeric digits. The test will also go to maximum brightness for a moment for comparison.

If the brightnesses aren't even, increase the test resistance to reduce the



Given the height of the enclosure we recommend, it's best to mount the Raspberry Pi as shown, so there's enough clearance. Note that the slim radio board I added down the left-hand side, with its antenna passing through a small hole in the rear panel.

Fig.6 (right): the wiring diagram for the Raspberry Pi Clock Radio.

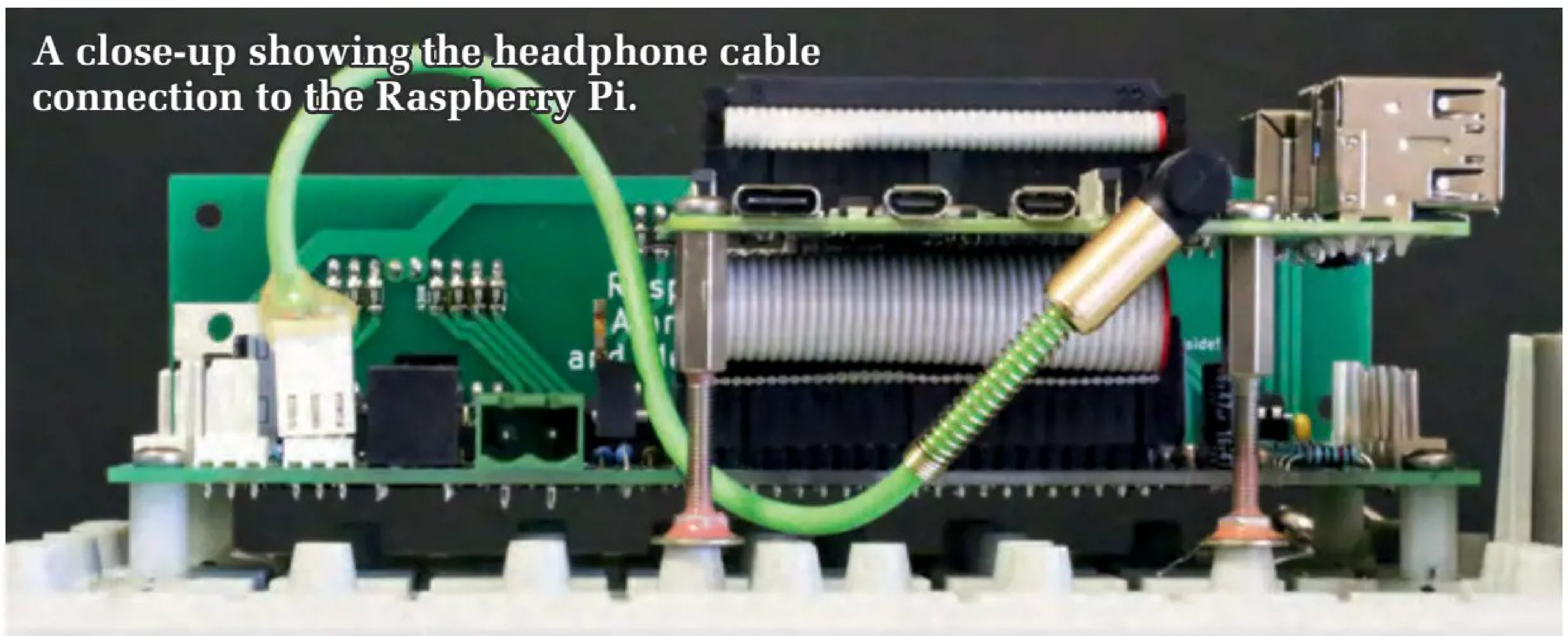
LED brightness or decrease the resistance to make it brighter. Always power down the clock before changing resistor values, as accidentally shorting out the current limiting resistor pads can cook your 7-segment display.

As you try different valued resistors, make step changes of about 25% in value, as you won't notice too much difference when trying smaller steps. For example, change from 430Ω to 560Ω to reduce the brightness or from 430Ω to 360Ω to increase it.

After determining the optimal resistances, you can source and install components for all the missing current limiting resistors, and your clock will be complete. If you don't want to wait, you could order 16 each of the values 360Ω, 430Ω, 560Ω, 750Ω and 1kΩ, as it's likely one of those will give a good match. Don't go lower than 360Ω to avoid exceeding the 74HCT4511 package current limit.

Testing

The proper functioning of each switch is most easily tested by enabling one of the software debug modes to generate a debug log. Do this in an SSH session; as described in the software installation documentation, the command is "sudo alarm-clock -B". As you press each button in the switch debug mode, a message will be logged, making it obvious that it is working.

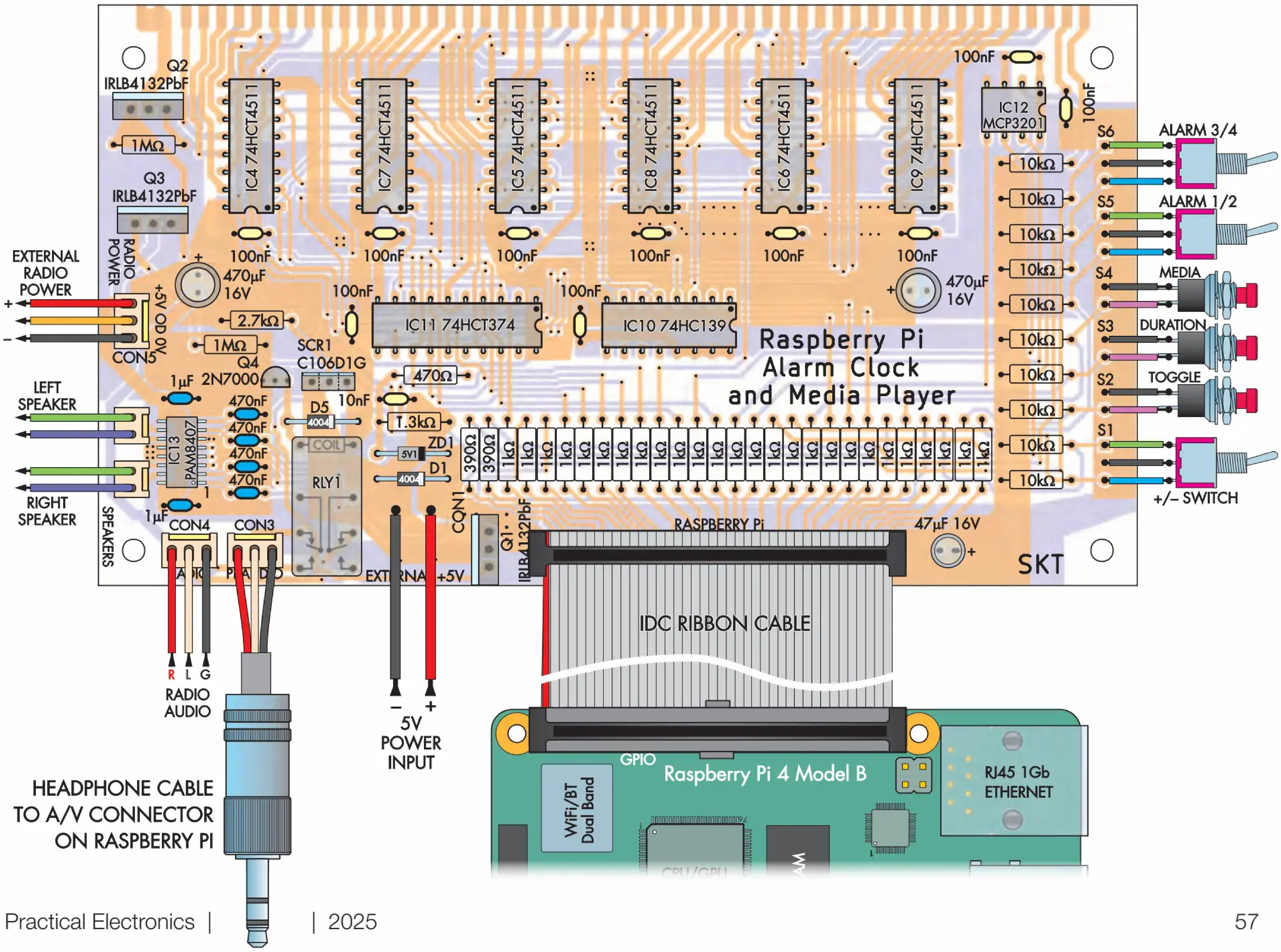

A close-up showing the headphone cable connection to the Raspberry Pi.

Assuming the LED displays are illuminated, if you cup your hand over the LDR (shielding its sides and the front surface) to reduce the light intake, you should see the display dim after a few seconds. LDRs can be rather sensitive to light, and the actual dimming behaviour will depend on what you've configured in the Clock settings.

You can use the command "sudo alarm-clock -X" to watch as the software measures the light level. You can verify that the numbers are smaller when the LDR is dark and become larger as the LDR is illuminated more.

Suppose you see this behaviour in the debug log, and there is a reasonable variation in the range between light and dark (for example, 500 counts or more). In that case, the circuit is working sufficiently well for you to be able to adjust the web sliders to achieve your desired dimming behaviour.

Use the "Start the display test" button on the web setup page to exercise each seven-segment display


Q2
IRLB4132PbF
1MΩ
Q3
IRLB4132PbF
IC4 74HCT4511
IC7 74HCT4511
IC5 74HCT4511
IC8 74HCT4511
IC6 74HCT4511
IC9 74HCT4511
IC12
MCP3201
100nF
S6
ALARM 3/4
S5
ALARM 1/2
S4
MEDIA
S3
DURATION
S2
TOGGLE
S1
+/– SWITCH
10kΩ
EXTERNAL
RADIO
POWER
RADIO
POWER
+
–
+5V
0V
CON5
470μF
16V
2.7kΩ
1MΩ
Q4
2N7000
SCR1
C106D1G
IC11 74HCT374
IC10 74HC139
Raspberry Pi
Alarm Clock
and Media Player
1μF
470nF
LEFT
SPEAKER
RIGHT
SPEAKER
IC13
PAM8407
D5
4004
10nF
470Ω
1.3kΩ
COIL
RLY1
5V1
ZD1
D1
390Ω
1kΩ
SPEAKERS
CON4
CON3
RADIO
PI AUDIO
EXTERNAL +5V
CON1
Q1
IRLB4132PbF
RASPBERRY Pi
47μF 16V
SKT
R L G
RADIO
AUDIO
– +
5V
POWER
INPUT
IDC RIBBON CABLE
HEADPHONE CABLE
TO A/V CONNECTOR
ON RASPBERRY PI
GPIO
Raspberry Pi 4 Model B
WiFi/BT
Dual Band
RJ45 1Gb
ETHERNET

and dots. If some LED segments never illuminate, check that the series current-limiting resistors are adequately soldered and that the legs of the 74HCT4511 chips are all correctly inserted into their sockets.

If just one of the six digits doesn't count in the correct sequence, the problem is likely with the associated 74HCT4511. If several of the six digits count incorrectly in the same way, check for problems with the GPIOs used for the data and address bus or potentially with the 40-conductor ribbon cable.

The last functions to test are the amplifier and audio inputs. Turn on the amplifier using the media player to stream local media or an internet radio site and verify that you're getting sound from the Raspberry Pi. If your speakers are dead, plug headphones directly into the Pi to confirm that it is generating audio; if so, you'll need to debug the amplifier section of the clock.

Back on the media player web page, if you type "radio" (without the quotes) rather than entering a path to a media file or URL, you should hear the relay click, and the audio will be taken from the radio input connector. The middle pin of CON5 will be pulled to ground via Q3 when the relay is energised so the radio module will be powered.

Putting it into a case

You will probably want to mount the clock into an instrument case, as I did with mine.

You probably shouldn't use a metal case for this project because it will interfere too much with the Raspberry Pi's WiFi and Bluetooth reception.

If you're capable of doing woodwork, I once saw a very cool clock idea that would turn this project into a talking piece. Timber veneer can be translucent if you shine a bright light through it from directly behind. The brightness of the clock's LEDs can be set quite high, especially the Lumex components, and they should shine clearly through a sheet of 0.4mm or thinner light-toned veneer.

You'd want to make yourself an MDF template for the front panel, possibly routing the top edge to round it off and drilling and filing out tightly fitting holes for the seven-segment displays. Mounting the front plane of the seven-segment displays immediately behind the veneer will provide mechanical backing to protect the veneer.

You would also need to drill a window for the LDR or mount the LDR elsewhere in the box. I'd love to see photos of this idea if it works as well as I expect.

If you choose the more conventional approach and use the plastic instrument case, I used a table saw and the finest toothed blade to cut a 3mm green Perspex sheet to the correct size to replace the front panel. Because the front panel mounting slot is 2mm wide, I firmly taped and then routed the edges of the Perspex so they fitted snugly.

2mm Perspex/acrylic is available, but I'm unsure if you can get the translucent green colour in that thickness.

If you feel it's necessary, you could mask and spray the inside of the Perspex bezel black so that only the displays show through, but I have not done this for my clocks and think they look fine.

I mounted my snooze button on the top of the case and all other buttons and switches, speaker posts and the power socket onto the rear panel that came with the case.

If you use the recommended instrument case which is 160mm deep, note that the clock board assembly with mounted LEDs amounts to just under 100mm of depth, and the shorter dimension of a Raspberry Pi is 56mm, not accounting for the 3.5mm audio plug and cable which protrude further. That means there isn't enough room to mount the Pi in the base behind the main board.

I mounted the Pi partially overhanging the main board, as shown in the photo on page 56. To raise it above the main board, you can use two long M3 screws through the base of the case, secured with a nut, and M3 standoffs fitted onto the tops of the long screws secured with weak Loctite (eg, 222).

Two short M3 screws will hold the Pi onto the M3 standoffs, while the two remaining Pi screw holes overhang the main board and are left vacant. It provides quite a solid mounting, provided you don't try to insert or remove the Pi's 40-pin expansion cable while the Pi is screwed down at only two points.

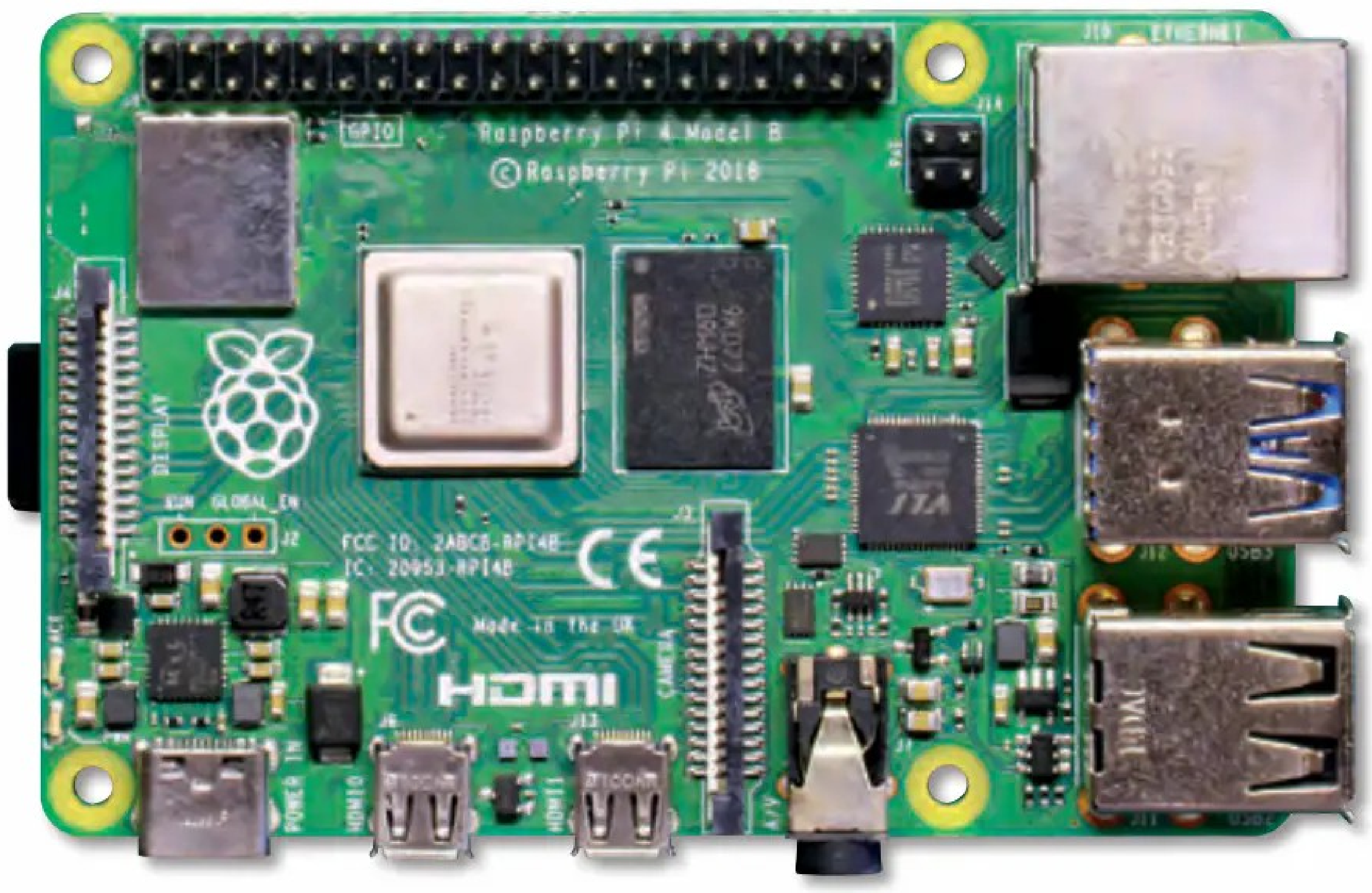
I found it sturdier to elevate the Pi using the moulded standoffs in the base of the instrument case, but the hole spacing wasn't quite right. I elongated one mounting hole to align with the Pi holes and secured the long screws with washers, nuts and Loctite 222.

Then with four standoffs attached to the main board and the perspex bezel in position, I carefully positioned the PCB and epoxied the standoffs to the instrument case, saving me the hassle of precisely measuring hole locations and drilling.

With the specified instrument case, there is still just enough room to squeeze a small radio board down one side if you'd like to include that option (as I did). To fit larger radio boards, you might need a larger case.

The case configuration I adopted has no external access to the USB or Ethernet ports. That wasn't a significant consideration for me, but if your case is sufficiently deep, you can rotate the Pi so that its network and USB ports are accessible through the back panel. In that case, you'll need to fold the 40 conductor ribbon into an L-shape to facilitate the connection between the main board and the Pi.

I specified a 150mm ribbon cable length and have successfully used that length with some of my prototypes. Shorter ribbons will be OK, but if you need to use a longer ribbon in yours,



While we used a Raspberry Pi model 4 to run our Clock Radio, you can also a model 2, 3, Zero W or Zero 2W.

be watchful for any problems when the LED display gets updated.

Suppose you see digit update errors when using a longer ribbon. In that case, you can slow down the settling and latching delay from its current default of 5µs by editing the "alarm_clock.c" source code file and changing the #defined symbol called WRITE_BYTE_DELAY to something larger than 5000 (nanoseconds).

The microprocessor on the Raspberry Pi consumes 1W, so expect the outside surface of the case in the vicinity of the Pi to be warmer than the room's ambient temperature. That should not be a concern, but you could add some ventilation to the case if you want to keep it cooler.

Using the software

The Clock Radio is designed to be intuitive once you've logged into the web interface, with plenty of help links to guide you. Still, if you want to tinker with it 'under the bonnet' or fully understand how it works, you'll want more details on how the software works.

We'll start with some of the more common features that most constructors will be interested in, then move on to the nitty-gritty of how the software works.

Configuring the Clock via the web interface

To reach the web interface, open a browser and go to http://clock.local or whatever system name or IP address you used to SSH into the Clock.

You will be greeted by the Clock's home page, which contains links to the various configuration and media player functions, a summary of the configured alarms, the playlist if media is currently playing, and a list of any other clocks found on the local network if any exist – see Screen 1.

Screen 1 shows other clocks that have been discovered on the network. If you only have one Clock, that boxed section will not be displayed.

Navigation should be reasonably intuitive. Use the Clock Setup page to configure your preferences. When you hover the pointer over some options, the browser will display hints and a description.

Creating alarms is a matter of filling in details from top to bottom on the Alarm Configuration page (Screen 2): what time, how long do you want it

Using the Clock as a Bluetooth speaker

Raspberry Pi models after the Pi 2 allow the Clock Radio to emulate Bluetooth speakers. Computers, phones and tablets can be paired with the Clock to play audio through its speakers. After 'pairing', devices only need to be 'connected' whenever you'd like to stream audio via the Clock's speakers.

A "Pair Bluetooth Device" icon on the Clock's web page allows devices to 'discover' it for pairing; you can also enable this mode by simultaneously pressing two of the clock's physical alarm selection switches. When activated, you'll have three minutes to complete pairing. You can pair as many devices with the Clock as you'd like.

The devices to be paired need to be in pairing/discovery mode at the same time, or they won't see each other; you'll need to carefully follow the documented pairing and connection process for your device. If you have difficulties pairing with the clock and are new to this process, confirm the procedure by pairing with something else.

When your device is connected, audio will be played through the Clock's speakers. Set the volume using either the physical volume up/down switch on the Clock or a slider on its web page.

Bluetooth audio stutters

The clock's Bluetooth streaming has been tested with Windows, macOS, Android phones and iPhones, and it has worked well using either the Pi 3 or the Pi 4. However, there are certain circumstances where you might notice occasional audio stutter and Bluetooth disconnections.

The most obvious situation is when connected Bluetooth devices move beyond their working range.

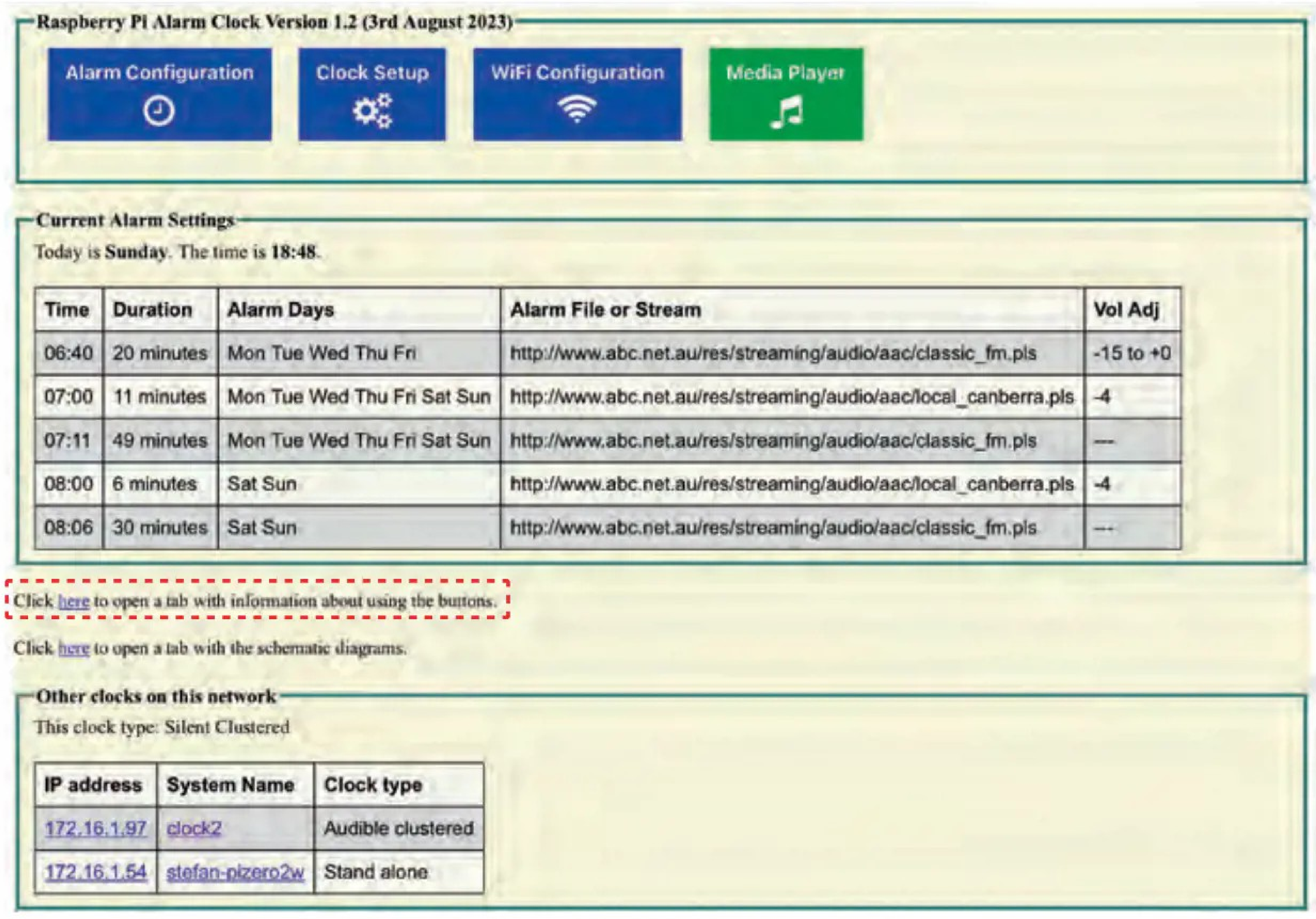
A less obvious reason for Bluetooth stuttering is WiFi interference, as explained in the panel last month. Therefore, you could experience audio stuttering if you access a Pi 3 based Clock's web interface at the same time as you're streaming Bluetooth audio to it.

Bluetooth channel assignment is typically statically allocated during the pairing process. If your Bluetooth channel experiences interference, you can try resetting pairing and starting over. There is a button on the web interface to reset all paired devices.

Suppose you are streaming Bluetooth audio when you start a media player stream on the clock, including when an alarm plays a media source other than the radio. In that case, the clock will disconnect the Bluetooth devices to play the media stream correctly. You will need to reconnect your Bluetooth device afterwards if you wish to continue streaming Bluetooth audio.

JOHN RAYMOND & REAL FEELS JOY RIDE
Be Still, My Soul
John Raymond
2:36 15
Infinity

The finished Clock Radio connected to an iPad via Bluetooth.



Raspberry Pi Alarm Clock Version 1.2 (3rd August 2023)

Current Alarm Settings

Today is **Sunday**. The time is **18:48**.

Time	Duration	Alarm Days	Alarm File or Stream	Vol Adj
06:40	20 minutes	Mon Tue Wed Thu Fri	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	-15 to +0
07:00	11 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4
07:11	49 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	---
08:00	6 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4
08:06	30 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	---

Click here to open a tab with information about using the buttons.

Click here to open a tab with the schematic diagrams.

Other clocks on this network

This clock type: Silent Clustered

IP address	System Name	Clock type
172.16.1.97	clock2	Audible clustered
172.16.1.54	stefan-pizero2w	Stand alone

Screen 1: the main web page shows the list of alarms and other Clock Radios on the network (if present). The buttons at the top are hopefully self-explanatory. Clicking on the link outlined in red provides information on what the six buttons/switches on the Clock Radio do. They can be used in combination, so quite a few functions are available.

to run, on what days of the week and should it recur indefinitely.

The alarm volume adjustment is relative to the current master volume, which is useful when chaining a sequence of alarms from different sources (where each source could have different audio levels). It can also be used to slowly increase or decrease volume with a succession of alarms.

Anywhere you see a text box for entering media, you can specify a streaming radio site URL, the full Linux path to a media file or playlist, or the word "radio" (without quotes) to use the external audio input. The path to media files on USB storage devices will normally start with "/media/".

Finding internet streaming sources is easy; any internet radio station that is accessible in your country and that can be listened to with a web browser will work with the Clock. However, determining the correct URL to use with the Clock can be tricky. Instructions on how to use a browser to discover the syntax for a streaming URL are included on the Clock's in-built web pages (also see Screen 2).

Chaining alarms

As I live in regional Australia, the national radio network I usually listen to inserts news bulletins from the nearest capital city. Sometimes, your ISP can allocate an IP address from further afield or the internet's routing changes, resulting in you hearing news bulletins from a completely different part of the country.

I realised I could use 'chained alarms' to work around this. When the hour ticks over and the local news commences, I configure a new alarm with its streaming media source being a local news radio station. I configured a new alarm to switch back to the original national stream after the news.

The chaining trick works well, and I've been enjoying local news by flipping streaming sources for several months.

When you chain alarms into a sequence, and a subsequent alarm trips before the timer for the earlier alarm has expired, the earlier alarm is replaced by the subsequent alarm. It will not come back even if the subsequent alarm completes. To return the first streaming source, you must chain a third alarm.

Waking up gently

You can use alarm chaining to be woken 'gently' by setting the volume adjustment low on the initial alarm, then chaining a series of subsequent alarms with progressively increasing volume. But there's a better way to achieve the same thing.

Each alarm includes a "target adjustment" setting as well as the "volume adjustment" setting. The volume adjustment is applied when the alarm first trips. Each minute after that, the digital volume control will make one step towards the target volume until it is reached (or the next chained alarm trips).

So, to gently increase volume when an alarm is tripped, you could set the volume adjustment to -15 and the target adjustment to +0. The alarm will start quietly and gradually get louder until it reaches the normal (system) level.

The volume adjustments are added to the current system-wide volume setting!

Adjusting the alarm(s)

Whenever you adjust the system-wide volume, the volume of all future alarms will adjust accordingly, as if you rotated the volume control knob on your traditional alarm clock and waited for the next alarm to trip.

If you've specified any non-zero volume adjustments for any alarm, those adjustments will be applied to the current system-wide volume to maintain the volume relativity.

The Clock Setup page has a "minimum alarm volume" setting. That ensures your alarms will still be loud enough to wake you if you've set a low system-wide volume.

Alarm-specific adjustments still apply to the minimum alarm volume to maintain relativity between different alarm levels and to ensure that adjustments continue to work as you expect. However, you won't be able to accidentally set the system-wide volume low and then sleep through an alarm. I therefore strongly recommend you check and adjust this setting during setup.

Using the buttons

Although there are only six buttons, they can be used in combination with each other, so you can do more with the buttons than you might think. The full description of button operation is provided in a link you will find on the Clock's internal web page, in Screen 1.

Because all functions can be reached via the web interface, it is not strictly necessary to build any or all of the buttons into the Clock. The choice is yours!

12/24 hour time display

The setup web page includes a configuration option for choosing between 12-hour and 24-hour displays. This refers to the LED display on the Clock itself, not how times are shown on the web pages. Times displayed in most web browsers are formatted according to the locale setting on the device running the web browser.

That means that although you may have configured one particular format on the web settings screen, your browser might steadfastly show a different format.

Playlist automatic resume

If internet streaming is not your thing, you can use a playlist of MP3 files from your own media library for your alarm. You can even chain alarms between your local playlist, the radio and internet streams.

When you initially create an alarm and specify a directory as an alarm source, the Clock's web server will build a playlist file of the underlying directory tree. Similarly, a playlist is created if you use the media player function and specify a directory.

When playing a playlist, the Clock remembers the last track and will commence from the following track when it uses the same playlist again. You can therefore chain from the playlist to the news and back again, and the playlist will continue from the track after where it left off.

Testing the hardware

When testing the hardware to ensure the switches are being recognised and ambient light levels are correctly measured, you'll need to use the Clock's software debugging mode to view the debugging log. This means you'll need to temporarily stop the operating system from managing the Clock.

By default, the "alarm-clock" program runs automatically at boot time. If it ever crashes, the operating system will restart it. Usually, its text output is hidden. To reveal and watch the log, you can run the alarm-clock program from within an SSH session and enable debugging.

To do this, within an SSH session, you can temporarily stop the alarm-clock program by issuing the command:

```
stop-alarm-clock
```

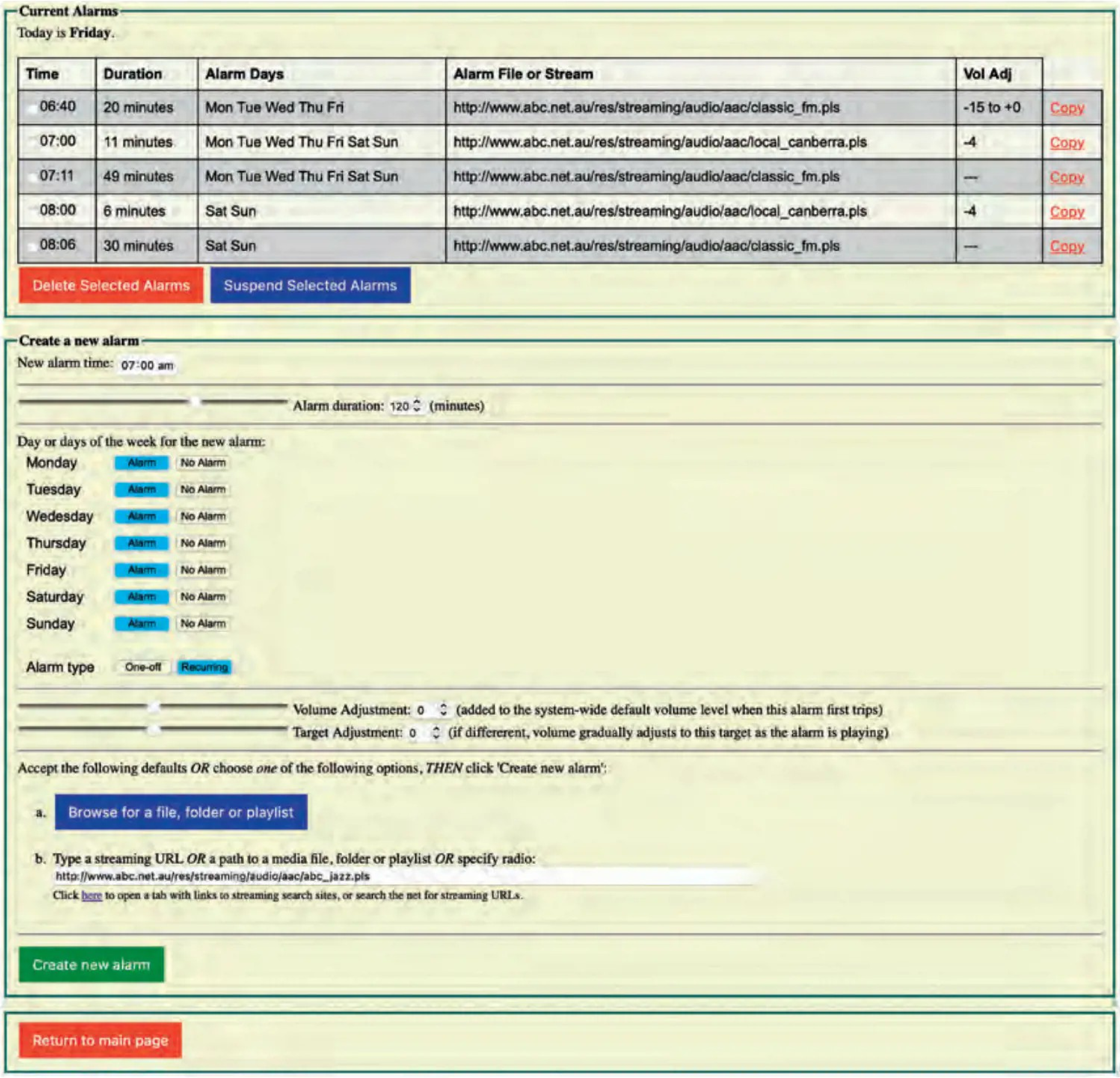
The Pi will revert to automatic clock management the next time you reboot.

Once the program has been stopped, you can run it manually with the command "sudo alarm-clock", or use "alarm-clock -h" to display the help options for the program. The command "sudo alarm-clock -V" will run the software with full logging. As it runs, messages and time stamps from different threads will intermix.

There's also a setting that enables logging to a file, for catching issues when you're not around to look. The Clock's setup file is located in "/etc/alarm-clock/setup.conf" and you can edit it to enable file logging using the following command, which will launch a text editor:

```
sudo nano
  /etc/alarm-clock/setup.conf
```

Look for the two following lines



Current Alarms

Today is **Friday**.

Time	Duration	Alarm Days	Alarm File or Stream	Vol Adj	
06:40	20 minutes	Mon Tue Wed Thu Fri	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	-15 to +0	Copy
07:00	11 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4	Copy
07:11	49 minutes	Mon Tue Wed Thu Fri Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	—	Copy
08:00	6 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/local_canberra.pls	-4	Copy
08:06	30 minutes	Sat Sun	http://www.abc.net.au/res/streaming/audio/aac/classic_fm.pls	—	Copy

Create a new alarm

New alarm time: 07:00 am

Alarm duration: 120 (minutes)

Day or days of the week for the new alarm:

Monday	Alarm	No Alarm
Tuesday	Alarm	No Alarm
Wedesday	Alarm	No Alarm
Thursday	Alarm	No Alarm
Friday	Alarm	No Alarm
Saturday	Alarm	No Alarm
Sunday	Alarm	No Alarm

Alarm type: One-off [✓ Recurring]

Volume Adjustment: 0 (added to the system-wide default volume level when this alarm first trips)

Target Adjustment: 0 (if differerent, volume gradually adjusts to this target as the alarm is playing)

Accept the following defaults *OR* choose *one* of the following options, *THEN* click 'Create new alarm':

a. Browse for a file, folder or playlist

b. Type a streaming URL *OR* a path to a media file, folder or playlist *OR* specify radio:

http://www.abc.net.au/res/streaming/audio/aac/abc_jazz.pls

Click here to open a tab with links to streaming search sites, or search the net for streaming URLs.

Screen 2: when configuring an alarm, you can choose the time, duration, which days it's active, whether it recurs, the volume adjustments and the audio source.

and remove the # symbols in the first column:

```
#VERBOSE=0x090
#LOG_FILE =
  "/var/log/alarm-clock.log"
```

Once you've made those changes, press CTRL+O to save them and CTRL+X to return to the command prompt.

For more information about the configuration, compiling and playing with the code, consult the readme files in the alarm-clock directory tree you extracted from the .tgz file.

Checking that the software is running

To check that the alarm-clock program is running, issue the following command:

```
ps -A | grep alarm
```

If it is running, you'll see a response showing how much CPU time it has used so far. If you see nothing, it is not running. Similarly, you can check for a running pigpio daemon (which must be running for us to control the Pi's digital I/O pins):

```
ps -A | grep pigpiod
```

We need a Bluetooth control process and its daemon to allow us to play Bluetooth audio, so there should be two items displayed when you run this command:

```
ps -A | grep bluetooth
```

The web server and its workers are usually waiting to receive connections, so there should be several items displayed with the command:

```
ps -A | grep apache
```

To see the complete list of all the running processes along with their memory consumption, type "top" and press Enter to start a self-refreshing display (type the letter q to exit).

The top command shows lots of helpful information, including how much CPU time each process is currently consuming, expressed as a percentage of a single CPU core.

As there are four CPU cores on most Pis, the Pi will be fully maxed out when the sum of all current CPU utilisation reaches 400%. The largest CPU hog is the pigpio daemon.

'Watchdog' reboot

During the testing and running of the four prototype clocks over the past six months, I saw two of the clocks lock up after a prolonged mains power brownout.

So, the software now incorporates several ways to determine that something has gone wrong and trigger a system reboot when it notices.

A hardware watchdog will reboot the Pi if the operating system fully locks up. If just one operating system task locks up, the Clock runs a half-hourly health check, triggering a reboot if something seems amiss. The web server also triggers software checks when somebody accesses a web page.

If you run "stop-alarm-clock" for debugging or testing, the health checks will eventually fail, and the automatic rebooting processes will reboot the Clock.

You can prevent these automatic reboots after you've issued the "stop-alarm-clock" command by restarting the alarm-clock program from the command line using the debugging flags you need.

The health-check features will not trigger a reboot if they see the alarm-clock program running.

Internet stream audio delays

You might notice something strange when using streaming internet radio stations for alarms. Suppose you set your alarm to trip precisely on the hour and expect the hourly news to wake you, as it would with a regular clock radio.

In that case, you'll discover that streaming radio stations can be anything from a few seconds to a few minutes behind the free-to-air version of the same radio station.

The news will never start exactly on the hour with an internet stream; it will always start just a little after!

A brief delay with internet streaming is to be expected, but I cannot explain why different radio stations stream with delays that change from day to day and month to month.

If you notice this happening with your preferred streams, there's nothing wrong with the Clock. It is just another artefact of the digital world we live in.

If you built a traditional radio into your Clock, over-the-air broadcasts will always arrive precisely on time through that medium. That is probably the only way to guarantee your news service commences with precision.

Enjoy your new clock, and if you build a couple, enjoy the liberation of both you and your partner being able to control them from either side of the bed!

Software updates

The Install_Clock script downloads and installs an automatic update client program called "unattended_upgrade" and configures it to check for updates at 3:30am every three days. If an update to any of the installed packages is released, such as a Linux security patch, the automatic updater will download and apply the change automatically.

If a reboot is required because of a security patch, that will be scheduled also without human intervention.

I have been running automatic updates on several Pis for years and have not struck trouble. However, if you prefer to run updates manually and at your own discretion, you can issue the following command over SSH to stop the automatic updates from occurring:

```
sudo systemctl disable unattended-upgrades
```

You can then manually look for and install updates using the command:

```
sudo apt-get upgrade
```

New versions of the Clock Radio software

After installing the clock software, there are two methods you could use to install an update, if one becomes available.

1. Open up the clock, remove the Pi's SD card, and plug it into your computer. Copy the new software .tgz file onto the SD card, put the SD card back into the Pi, exact the file contents and then rerun the Install_Clock.sh script.
2. Copy the new file into the root of your home account on the Pi over the network, using the Pi's Samba file server.

The following commands can be used to install the new software, assuming you copied the .tgz file into your Pi's home directory (assuming the update is v02):

```
tar zxf alarm-clock_v02.tgz
cd alarm-clock
make
make install
```

Practical Electronics

UK readers SAVE £1 on every issue

SUBSCRIBE NOW!

01202 087631

Take out a one-year subscription and save more than £10 over the year.
Even better – save over £1 per issue if you subscribe for two years – a total saving of £26.

Overseas rates represent exceptional value

You also:
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if your local newsagent sells out**

Order by:
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £33.99, 12-months £59.99, 24-months £114.99

Europe Airmail: 6-months £38.99, 12-months £69.99, 24-months £129.99

Rest Of The World Airmail: 6-months £46.99, 12-months £84.99, 24-months £164.99

Cheques (in **£ sterling only**) payable to *Practical Electronics* and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

Also via our online shop at**: www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £29.99 for 12 months, more details at: **www.electronpublishing.com**

SUBSCRIPTION ORDER FORM

☐ **6 Months: UK £33.99, Europe £38.99 (Airmail), Rest of the World £46.99 (Airmail)**

☐ **1 Year: UK £59.99, Europe £69.99 (Airmail), Rest of the World £84.99 (Airmail)**

☐ **2 Years: UK £114.99, Europe £129.99 (Airmail), Rest of the World £164.99 (Airmail)**

To: **Practical Electronics Subscriptions**
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

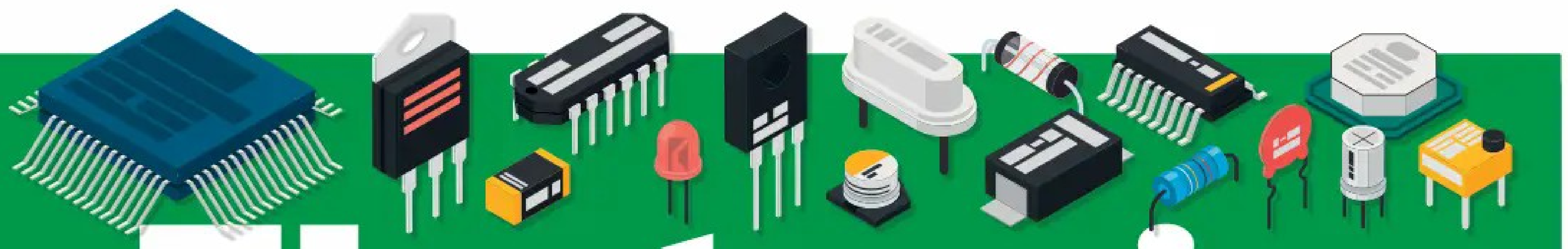
Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

Part 1: Introduction



Precision Electronics

This is the first article in a series covering the basics of precision electronics design. The practical series will cover a range of topics, including precision op amps, instrumentation amplifiers, signal switching and noise. I will use real examples and real components to demonstrate the concepts.

By Andrew Levido

While I aim to cover this topic from a practical perspective rather than a theoretical one, some theory is unavoidable. Along with explaining the concepts, I hope to give a few tips and tricks along the way. Since most devices built today include a microcontroller, we will also look at analog-to-digital and digital-to-analog conversion.

What is precision?

We should start by defining 'precision' in the context of precision circuits. We should also distinguish between precision and accuracy, two often confused terms.

Both precision and accuracy are ways of looking at the error in the measurement of a physical or electrical quantity. Accuracy describes how closely a measurement or series of measurements matches the 'true' value. In practice, that more likely means how closely it matches an accepted proxy for the quantity, probably traceable to some international standard.

Precision describes how closely a series of measurements match each other. It relates to the repeatability of a measurement – how confident we can be that another measurement in one minute, tomorrow, or next year will be the same as the one taken now.

Alternatively, it could indicate how confident we are that the measurement taken by the second, 100th or 10,000th unit off the production line will perform identically to the first one.

Fig.1 illustrates this nicely. It is a histogram of 16 different measurements of a nominal 10.0V source taken over time. The mean of the samples is 9.9V, with a spread of ±0.2V around this (from 9.7V to 10.1V). The mean differs from the 'true' 10V value by 0.1V.

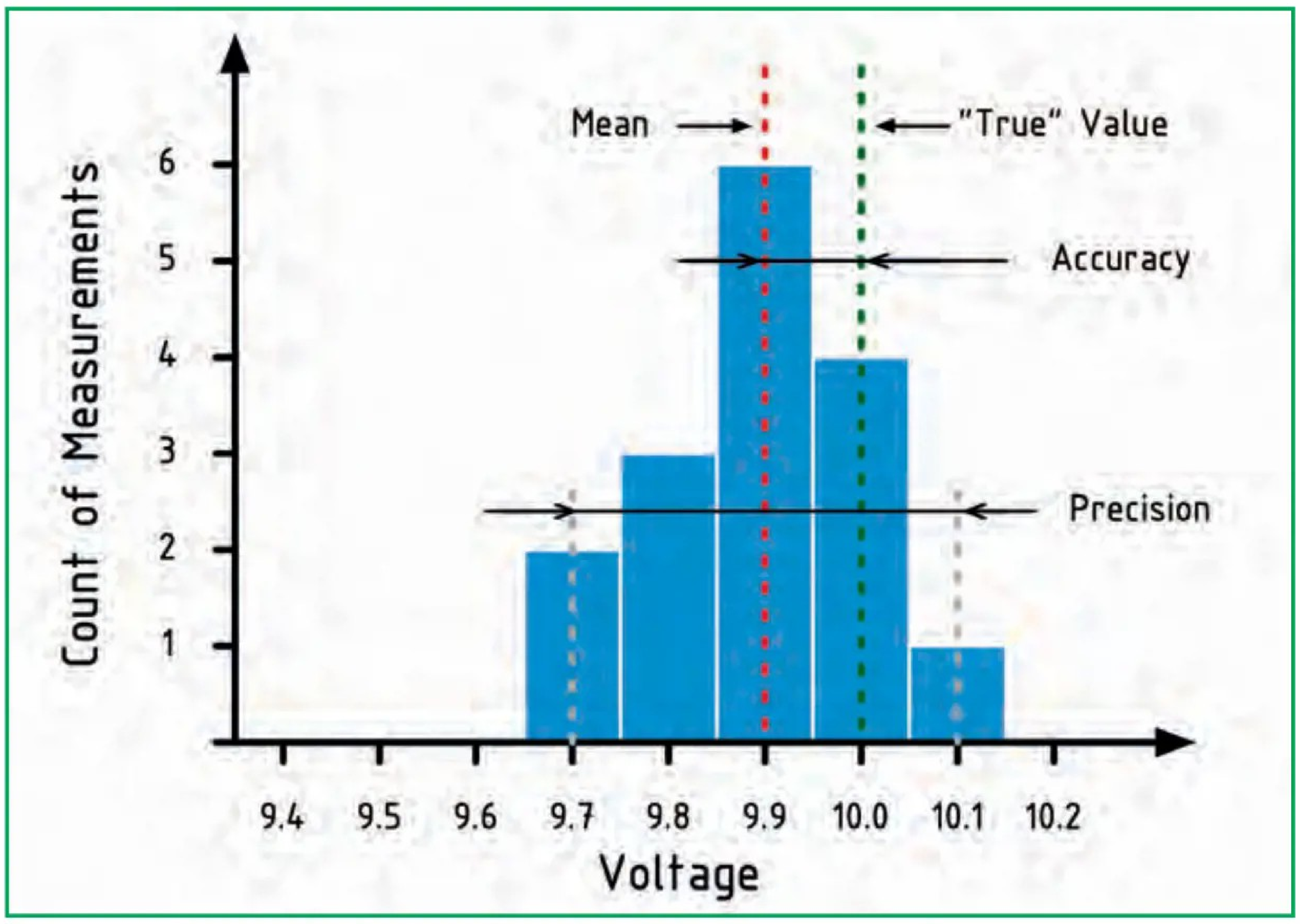


Fig.1: 16 samples of a nominally 10V source. The measurement accuracy is the difference between the sample mean and the 'true' value of 10V, while the precision is the spread of samples about the mean. Here, the precision is ±0.2V (absolute) or ±2% (relative).

Therefore, we can say that our accuracy is within ±0.1V of 10.0V or ±1%. The precision of our measurement is ±0.2V around the 9.9V mean, so within ±2%.

Precision and accuracy are related but independent quantities. We can have precision without accuracy and accuracy without precision (although the latter would be of limited value). Note that in the example above, an accuracy of ±1% does not mean that every measurement will be within ±1% of the actual value since the measurement precision is not good enough to allow that.

Accuracy is all about traceability and calibration, whereas precision is all about understanding and controlling the sources of uncertainty or error in our circuits. It is not always about achieving the highest levels of precision – it is about getting 'good enough' results for the application, which requires us to know what the precision of our circuit is.

From the example above, you will have seen that we talk about precision in both absolute terms, such as ±0.2V, or in relative terms using percentages (±2%). We also use parts per million (ppm) for relative precision when the numbers get very small; for example, 0.01% equals 100ppm. If we have extremely good precision, we might even talk about parts per billion (ppb)!

We can always measure the precision of a circuit after it is built, but we have just seen that one sample isn't enough. Also, we usually want to be sure our design will meet the precision targets before we commit to mass manufacture. Precision circuit design is the process of keeping careful track of errors and uncertainties and how they accumulate to impact the overall precision of the circuit of interest.

Sources of uncertainty

Before we get into a practical example, it might help to understand where these errors and uncertainties come from. Many errors result from complex interactions of various causes, but it helps to think of them in some broad categories:

Limitations of physics

Real-world limitations introduce errors. For example, there is no such thing as a perfect insulator, so leakage currents occur. It is impossible to source or sink infinite current, so devices must have some finite output impedance, which means outputs will change with load.

Noise

Another inescapable result of physics is the electrical noise caused by the random movement of electrical charges in certain materials. This can significantly impact measurements involving small quantities (microvolts and microamps, or even nanovolts and nanoamps!). Noise is a whole topic in itself that we will cover later in this series.

Temperature

Sadly, almost everything in electronics changes with temperature, and usually not for the better. Resistor values change, noise increases and offsets drift. The wider the temperature range your device will be subject to, the more this will be a problem you must address.

Frequency and time

Like temperature, frequency changes almost everything. A parameter specified at DC may vary considerably as frequency increases. Some things get worse over time, too. MLCC capacitors lose capacitance with age, and even the frequency of crystals can drift over time. It's not the biggest problem you will likely encounter, but it is worth being aware of.

Manufacturing variation

Even a well-designed component, using the best materials and a good manufacturing process, will have some degree of variation between parts. It is impossible to make them all absolutely identical. Common examples include resistor tolerances and op amp input offset voltages. There will be a natural spread of these values around a mean (the nominal resistance for resistors or 0mV for op amp offset voltages).

Understanding component limitations

There are no perfect components, just as there are no perfect circuits. Optimising for one parameter may have a detrimental effect on another.

One example that springs to mind is the common multi-layer ceramic capacitor (MLCC). Many of these use a dielectric material that allows the manufacturers to cram a huge amount of capacitance into a tiny volume for a ridiculously low price. The downside is that the capacitance is highly sensitive to temperature, applied voltage and ageing.

The component variation with these conditions can easily be two or three times the nominal tolerance of the capacitance. That is the price you pay for 10¢ 10µF 0402-size capacitors.

Sticking with the example of ceramic capacitors, do you know what it means when a capacitor is labelled X7R, X5R, Y5V, C0G, NP0 etc? It is related to the temperature range and how much the capacitance varies over it, but it is actually much more than that. For example, these codes also affect how capacitance changes with voltage. This shows why it pays to do your homework!

Manufacturers are not always as forthcoming about a part's limitations as they are about its features (especially on the front page of the data sheet). Be wary of typical values compared to worst-case values. You must read the data sheets carefully and thoroughly.

Don't just read the data tables – often, the graphs give useful information about how a device will perform that is quite different from the flattering conditions under which the nominal values are derived.

A practical example

Despite all this, it is, of course, possible to design high-precision circuits, and there are a few handy tricks that can help us get there. To get started, we will use a simple example that we can build upon in subsequent articles.

Imagine we are designing a DC power supply to power a microcontroller-based circuit. We want to measure the current consumed by our device over the range of 0A to 1A. We would ultimately like to measure currents down to the microamp level (or lower) if possible, since our device may go into sleep mode.

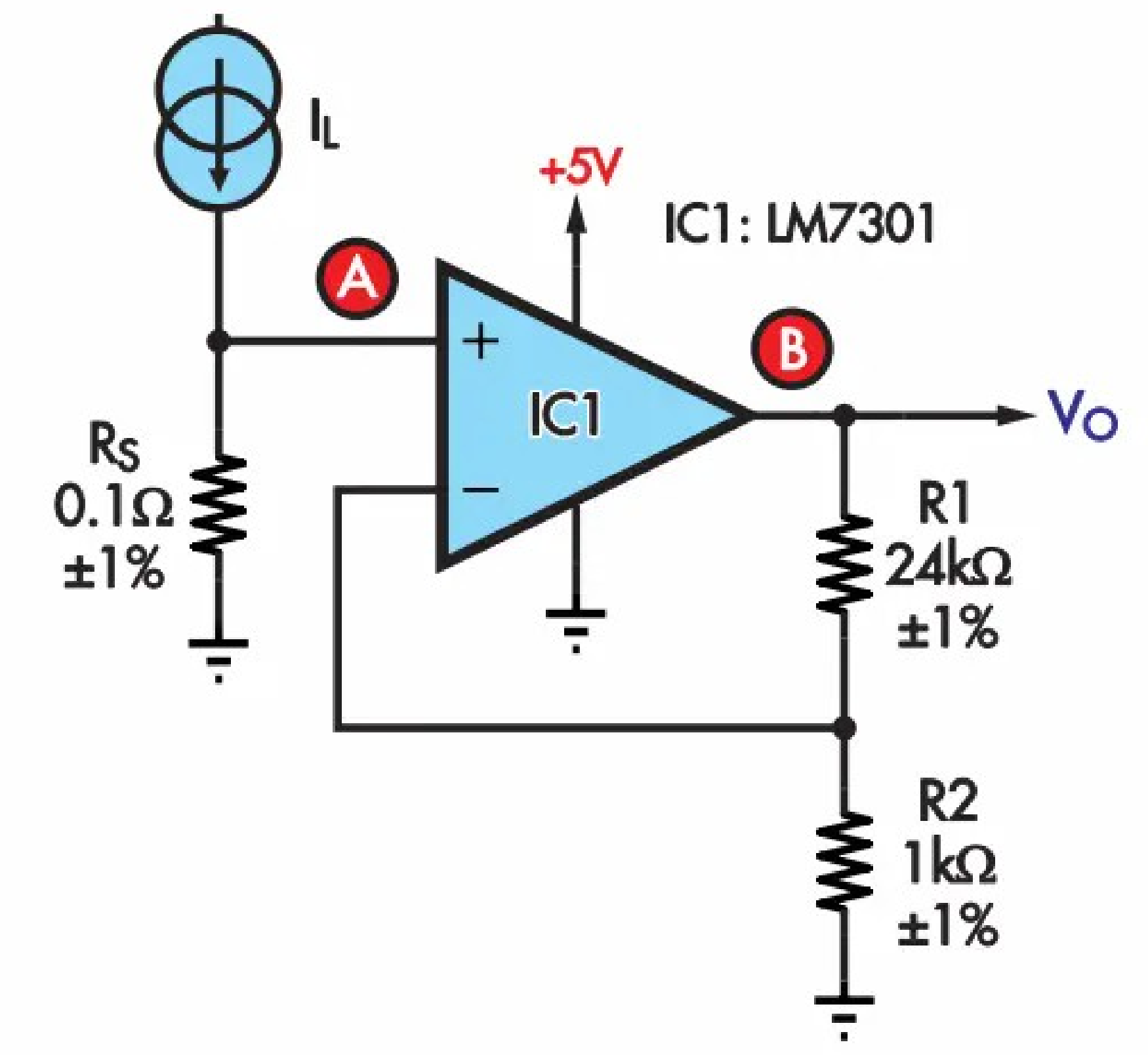


Fig.2: our first attempt at a current-measuring circuit. The 0-1A current to be measured (I_L) flows through R_S and the resulting voltage is amplified by IC1 to produce a 0-2.5V output. It uses regular 1% resistors and a low-cost rail-to-rail op amp.

This isn't easy to achieve. We will develop the idea over the next few articles, but let's start by working out what sort of performance is possible with some very basic components and a straightforward circuit. Fig.2 shows the circuit we will begin with.

On the left is a 0.1Ω resistor used as a current shunt. For the time being, we will assume it is ground-referenced. This shunt will drop 100mV across it at the full 1A load. We need to amplify this signal to get it into the range of an analog-to-digital converter, say to around 2.5V, which means we need an amplifier gain of 25.

I have used a low-cost general-purpose rail-to-rail input and output (RRIO) op amp, the LM7301, to start with since its inputs and outputs can swing to the rails. We'll also use standard 1% tolerance resistors to set the gain. Initially, we will power this part of the circuit with a single 5V supply.

To estimate the precision that we can expect from this circuit, we need to move through the circuit one element at a time, find its contribution to the overall error and sum them somehow. We will take this very slowly initially to illustrate the process.

At node A, we will see a voltage proportional to the load current but with some uncertainty due to the resistor

Parameter	Test Conditions	TYP	MAX
V_{OS} – input offset voltage	T_A = 25°C	0.03mV	6mV
	$T_A = T_J$	N/A	8mV
TCV_{OS} – input offset voltage average drift	$T_A = T_J$	2µV/°C	N/A

Fig.3: this extract from the LM7301 data sheet shows the expected input offset voltage. At 25°C, it is specified to be ±30µV (typical) and ±6mV (maximum) – quite a range! I suggest using the latter in your designs.

Adding two quantities with errors:

$(z + \Delta z) = (x + \Delta x) + (y + \Delta y) = (x + y) + (\Delta x + \Delta y)$

$\rightarrow z = x + y, \Delta z = \Delta x + \Delta y$

Multiplying two quantities with errors:

$(z + \Delta z) = (x + \Delta x)\bullet(y + \Delta y) = x\bullet y + x\bullet\Delta y + y\bullet\Delta x + \Delta x\bullet\Delta y$

$\rightarrow z = x\bullet y$

$\Delta z \approx x\bullet\Delta y + y\bullet\Delta x$ *and* $\frac{\Delta z}{z} \approx \frac{\Delta x}{x} + \frac{\Delta y}{y}$

Fig.4: when adding or subtracting quantities with uncertainties, the uncertainty of the result is the sum of the absolute uncertainties, shown at the top. When multiplying or dividing, the uncertainty of the result is approximated by the sum of relative uncertainties, shown below.

Measured Data		Error	
Current	**V_{OUT}**	**Abs.**	**Rel.**
0.0	25.0	25.0	1.0%
99.7	251.9	2.7	0.1%
199.8	515.2	15.7	0.6%
299.7	769.6	20.4	0.8%
399.9	1021.3	21.6	0.9%
499.9	1272.5	22.8	0.9%
599.9	1523.9	24.2	1.0%
699.9	1777.0	27.3	1.1%
800.0	2030.1	30.1	1.2%
900.0	2282.1	32.1	1.3%
1000.0	2533.3	33.3	1.3%

Table 1 – measured results from the Fig.2 circuit using a single supply (+5V). ***Units: Current (mA), V_{OUT} (mV), Absolute (mV), Relative (%).***

tolerance. The resistor tolerance is 1%, so it will have an absolute resistance value of 100±1mΩ. We will therefore see a voltage across it of 100±1mV at full load.

We will also see the op amp's input offset voltage appearing at node A. Fig.3 shows the relevant extract from the LM7301 data sheet. The input offset voltage at 25°C is specified to be ±30µV (typical) and ±6mV (maximum). The maximum offset is more than 100 times the typical figure! We will use the worst-case value for reasons I will discuss below.

We now have two quantities (voltage across the resistor and the op amp offset voltage), each with its own uncertainty, that we need to sum. The error in the total value will simply be the sum of the absolute errors of each part. This probably seems obvious, but you can see the maths that proves it in Fig.4.

That figure also shows the less obvious result: that the total error when two quantities are multiplied is approximated by the sum of the relative errors of each quantity. The approximation works because we can ignore the Δx•Δy term if the errors are small.

This leads to an important rule for precision circuit design: If adding or subtracting quantities, sum the absolute errors; if multiplying or dividing, sum the relative errors.

So, back to our circuit. Summing the absolute errors at node A gives a total error of ±7mV. You can probably already see this is a potential problem (no pun intended), but let's keep going. At node B, we will see the voltage at node A multiplied by the gain of the op amp stage. The gain with two 1% gain-setting resistors will be 25±2%, or 25±0.5 in absolute terms.

The total error at the circuit output (Node B) will therefore be the sum of the relative errors of the Node A voltage (±7%) and the gain (±2%), or ±9%. This corresponds to about ±225mV absolute error in the 2.5V full-scale signal. Clearly, that is not acceptable. The op amp offset voltage is the biggest contributor by far and is pretty easy to deal with. But how will this circuit perform in real life?

Practical results

I built this circuit and measured the results shown in Table 1. You won't be surprised that they are much better than the worst-case estimate of ±9%. This is because the errors result from statistical variation, and there is a much higher probability that any given

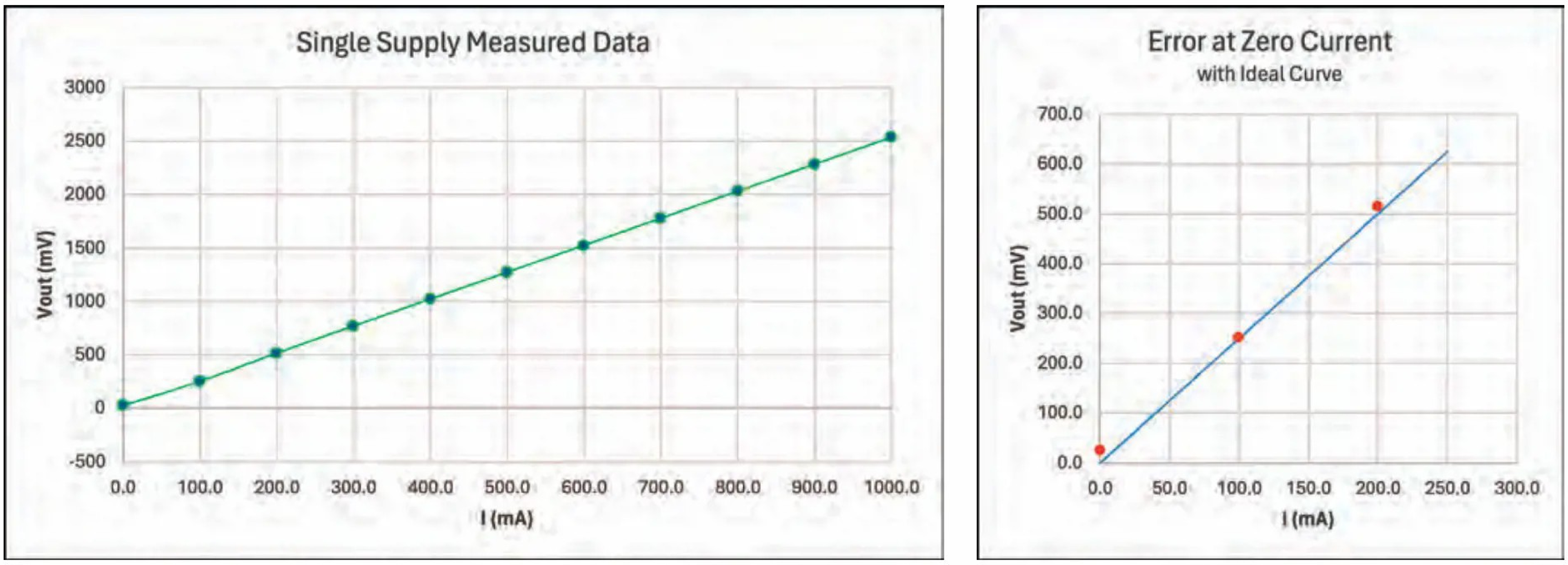


Fig.5: at left is a plot of the measured results from the Fig.2 circuit; note the subtle kink in the curve near zero. The close-up on the right clearly shows that the output is too high at 0A due to the op amp's limited output swing.

Measured Data		Error	
Current	V_{OUT}	Abs.	Rel.
0.0	-41.5	-41.5	-1.7%
97.9	203.7	-41.1	-1.6%
198.2	454.6	-41.9	-1.6%
298.3	693.3	-52.5	-2.1%
398.3	944.1	-51.7	-2.1%
498.3	1197.2	-48.6	-1.9%
598.3	1447.5	-48.3	-1.9%
698.0	1728.3	-16.7	-0.7%
798.0	1982.2	-212.8	-0.5%
898.0	2235.2	-9.8	-0.4%
998.0	2488.8	-6.2	-0.2%

Table 2 – raw results from the Fig.6 circuit with a dual supply (±5V).

Measured Data		Error	
Current	V_{OUT}	Abs.	Rel.
0.0	12.8	12.8	0.5%
97.9	253.9	9.2	0.4%
198.2	500.7	5.2	0.2%
298.3	735.5	-10.3	-0.4%
398.3	982.1	-13.6	-0.5%
498.3	1231.1	-14.7	-0.6%
598.3	1477.2	-18.5	-0.7%
698.0	1753.4	8.4	0.3%
798.0	2003.1	8.1	0.3%
898.0	2252.0	7.0	0.3%
998.0	2501.4	6.4	0.3%

Table 3 – the Table 2 data after applying a fixed offset and gain corrections.

sample will be near the mean or nominal value than an outlier.

The full-scale error was 33mV, or 1.3%, and the errors reduce at lower currents except at the bottom of the range, where there seems to be some kind of anomaly.

You can see this also in the plot of the results in Fig.5, on the left. The full set of results looks OK except for the zero-current reading, which is slightly off. The first three readings, along with the ideal response, are shown on the 'zoomed in' plot on the right of Fig.5. There is clearly a problem at or near zero current.

We know the op amp offset voltage is not causing this, because that would appear as a consistent vertical shift of the measurements above or below the ideal line. It is not caused by gain error, because that would appear as a variation in the slope compared to the ideal line. Something else is going on – there is a small but definite 'bend' in the measured results at the bottom end.

The culprit is the op amp's output swing. While the LM7301 claims to be a "rail-to-rail" output op amp, a close look at the data reveals that with a 5V supply and a 10kΩ load, the output typically won't go below 70mV (and isn't guaranteed to go below 120mV). We are measuring 25mV, which is better than claimed. This is a very good swing, better than most op amps, but it isn't rail-to-rail as advertised!

We would prefer to avoid such nonlinearities because they are harder to deal with than purely linear errors such as fixed offsets or gain errors, as we shall see. I refined my circuit by adding a negative supply rail (Fig.6). Running the tests again produced the data shown in Table 2 and plotted in Fig.7.

In some ways, this looks worse than our first test! The most significant error is just over −52mV or 2.1% of full scale. This error occurred mid-scale, with the absolute error at zero being −42mV; at full scale, it is only −6mV (0.2%).

The good news is that the points are fairly linear. The dotted line in Fig.7 is a line of best fit, using the equation shown on the graph. This line suggests there is a fixed offset error of −54.5mV and a gain error (the difference between the slope of the line and the ideal slope of 2.5) of about 1.7%.

The fixed error comes mainly from the op amp's offset voltage, which must be around −2.2mV (taking the gain of 25 into account). The gain error comes largely from the resistor tolerances. The good news is that there is no longer a bend in the plot.

Note that the op amp offset is less than the quoted worst-case figure (±6mV), but by no means does it fall within the typical figure of ±30μV. This is just one sample, but it does illustrate the danger of assuming your results will match the 'typical' figures in the data sheet.

We will improve this result next time by selecting a 'better' op amp and tighter tolerance resistors. But just for a moment, let's look at another solution. We could compensate for both of these errors (offset & gain) by adding a fixed correction – either through analog trimming or, more likely these days, in software on the microcontroller.

Just because we can, let's look at how much we could improve these readings by applying gain and offset correction using the values from the

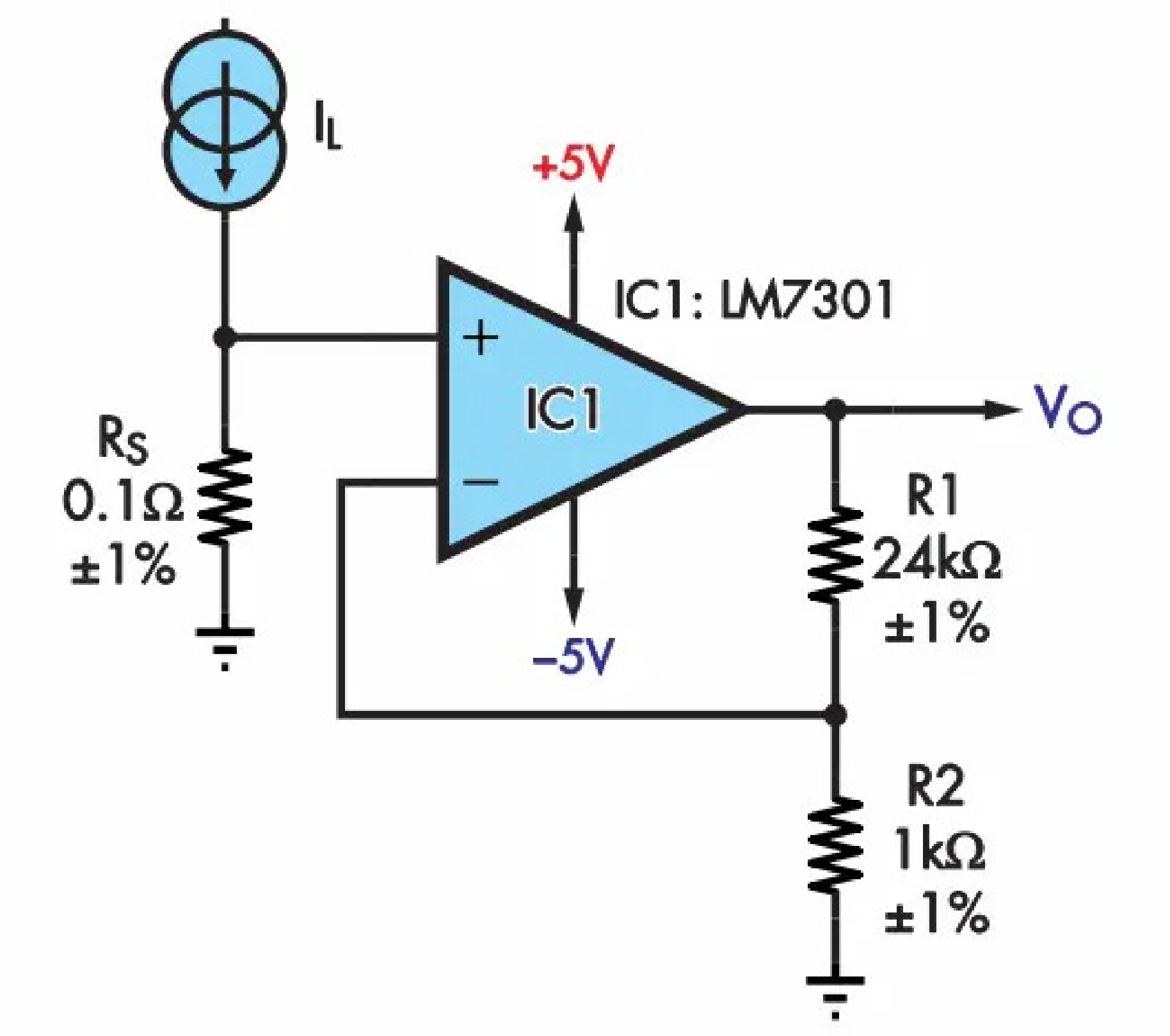


Fig.6: powering the op amp from dual supply rails (±5V) fixes its output swing problem. Otherwise, this circuit is identical to Fig.2.

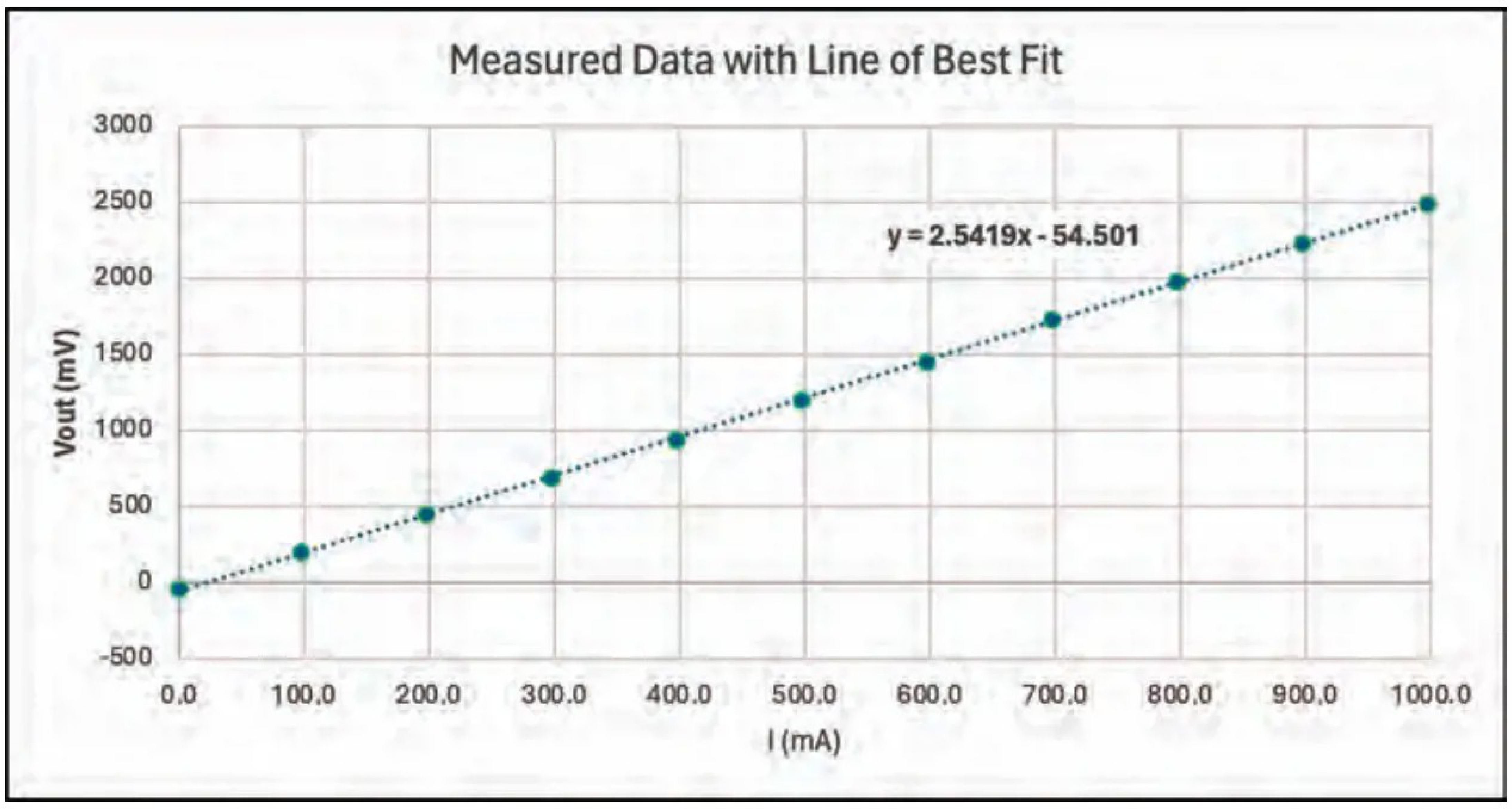


Fig.7 (right): the measured result of the Fig.6 circuit, along with a calculated line of best fit (dotted). There is now a fixed offset and gain error that can be trimmed out in either the analog or digital domains.

line of best fit. Table 3 shows the corrected results. Now the absolute error is never worse than about ±20mV, or 0.75% of full scale. Not bad, given the parts we have chosen.

This is one of the big secrets of precision design. You can usually trim out fixed offset or gain errors to some significant degree. The emphasis should be on the word "fixed". It's way more difficult to trim out non-linearities or errors that change over time, such as temperature drift.

Temperature effects

To examine the effect of temperature, I want to introduce the idea of the error budget table. This is just a way of capturing the uncertainties we discussed above in a neat tabular form. Table 4 shows an example. You can use any format you like, but this is how I generally do it.

Under the "At Nominal 25°C" section, you will see each step we went through in the above example, capturing the nominal value and relative and/or absolute uncertainty.

For example, Line 1 is the shunt resistor and Line 3 is the op amp offset. Lines 2 and 4 are calculated values and are shown in bold text. I always show both the absolute and relative errors on calculated lines. At Line 8, we get to the ±225mV and ±9% error figures calculated above.

The second part of the table brings the temperature-dependent errors into the picture. We obviously have to know the temperature range of interest to calculate these uncertainties. I have chosen a range of 0°C to 50°C (±25°C either side of the nominal 25°C) in this example.

The data sheet for the shunt resistor I used (Stackpole CSR1225) tells me that its temperature coefficient (tempco) is 100ppm/°C. This means we will see a resistance change of up to ±2500ppm or ±0.25% over the range of interest on top of the 1% tolerance.

Similarly, the op amp's offset voltage has a drift of ±2µV/°C, corresponding to ±50µV. This is already more than the ±30µV 'typical' offset at 25°C claimed in the data – another reason to take 'typical' values with a grain of salt.

If we continue with the rest of the analysis in the same way, we arrive at a variation of about ±0.8% over the proposed operating temperature range. Even if we could trim out all of the 25°C error in software, we are left with a temperature-dependent error approaching 1%. We will look at how we can reduce this in further instalments.

Optimist or pessimist?

One objection that frequently comes up when we are summing worst-case errors in this way is that we are being overly pessimistic in our design. We are assuming that errors will accumulate in the worst possible way. For example, we have assumed that our gain error is 2%, which would only be the case when both gain-setting resistors are at the extremes of their tolerances and in opposite directions.

If they were both high or low by the same percentage, this would cancel out, and the gain would be unaffected.

Is it reasonable to take this pessimistic view? What if our circuit had 10 gain-setting resistors instead of two? Would it be reasonable to assume they would all be at their tolerance extremes in the worst way? There is no correct answer to the question, but I can suggest some guidelines.

Uncertainty is a statistical game – it's all about probabilities and consequences. If the likelihood of the worst case occurring is low and its consequences are not severe, it is probably OK to make some concessions.

But if the probability of an error occurring is high (eg, if you are making a lot of something), or the consequences of any errors are significant (dangerous, expensive or embarrassing), a cautious approach is better.

One concession you might choose to make is to assume that the sources of error are uncorrelated. In such cases, it is possible to add errors (absolute or relative) as the root sum of squares. In our example of 10 gain-setting resistors, each with a 1% tolerance, we would come up with a gain error of ±3.1% instead of 10%.

But I urge caution. The root sum of squares is just another statistical tool – it works best when there are a great many samples in a truly random and uncorrelated distribution. We do use this type of summation for noise, which fits these criteria, as we shall see in a later article.

Remember that if some resistors have the same value, they will likely come from the same batch. In fact, they will probably have been manufactured sequentially. So they will very likely be off by roughly the same amount and in the same direction. In other words, the errors won't be uncorrelated at all!

In some cases, that can help you; eg, if you're relying on matched resistor values. Still, you must examine the specific circuit to determine whether correlated errors will help or hurt your precision.

Summary

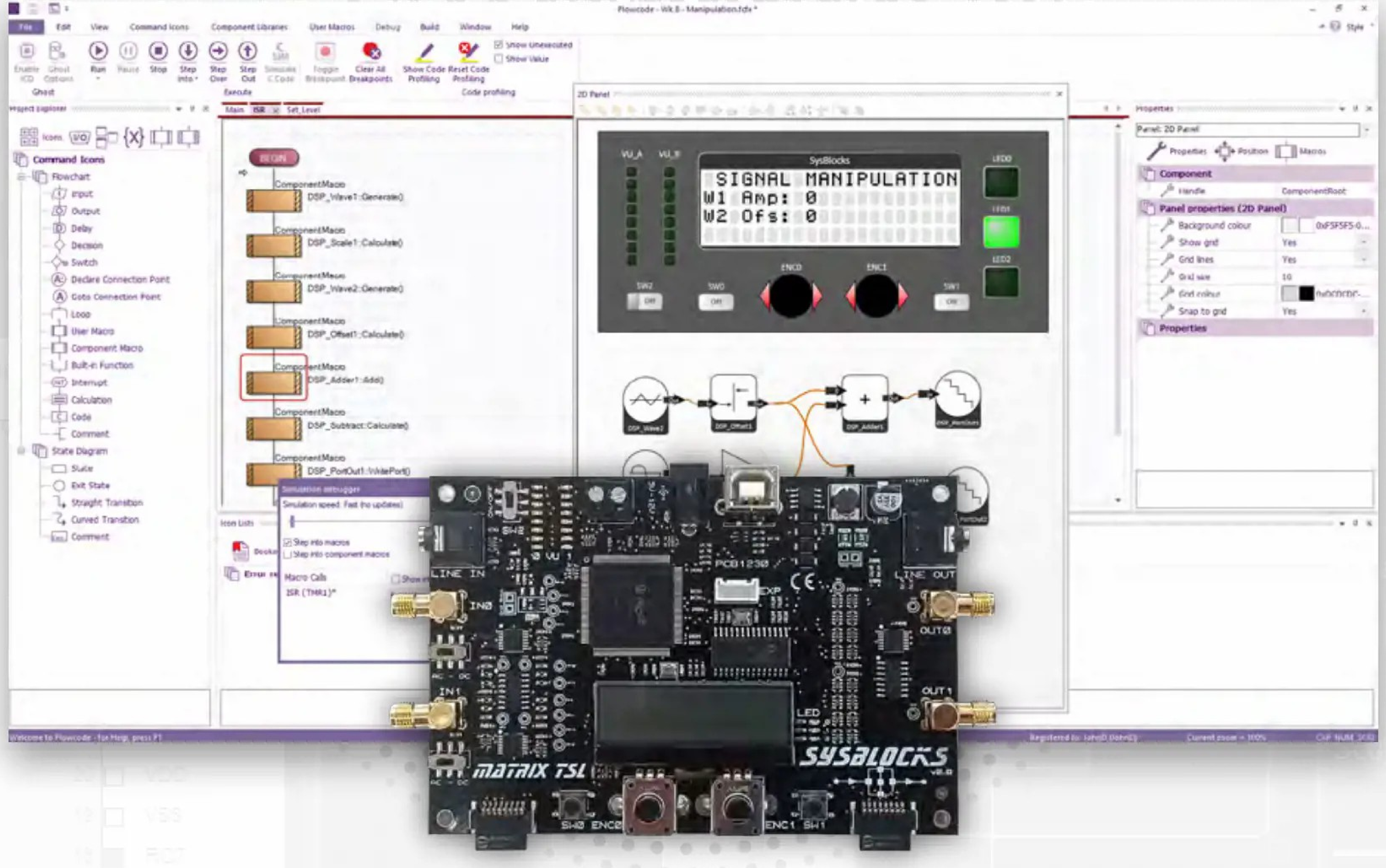
At this stage, it has become clear that our simple circuit is probably not up to the job of monitoring the current in our supply if we want anything better than a couple of percent resolution.

We can trim out the worst of the ±9% error down to a little better than 1%, but we will have another 1% or so of error over the temperature range. This 2% error means a ±20mA uncertainty. We'll have to do better next time!

Table 4: Error Budget Table for our Application

		At Nominal 25°C		0-50°C (Nominal ±25°C)	
Error	**Nominal Value**	**Abs. Error**	**Rel. Error**	**Abs. Error**	**Rel. Error**
Shunt Resistor: Stackpole CSR1225 (1% 100ppm/°C)	100mΩ		1.00%		0.25%
Node A Voltage due to I × R shunt	**100mV**	**1mV**	**1.00%**	**0.25mV**	**0.25%**
Op Amp: LM7301 (V_{OS} ±6mV, 2µV/°C)	0mV	6mV		0.05mV	
Node A Voltage total (Line 2 + Line 3)	**100mV**	**7mV**	**7.00%**	**0.3mV**	**0.30%**
Op Amp Gain Resistor R1: Yageo RC0805 (1% 100ppm/°C)	1kΩ		1.00%		0.25%
Op Amp Gain Resistor R2: Yageo RC0805 (1% 100ppm/°C)	24kΩ		1.00%		0.25%
Op Amp Gain (R1 + R2) ÷ R1	**25**	**0.5**	**2.00%**	**0.125**	**0.50%**
V_{OUT} (Line 4 × Line 7)	**2.5V**	**0.225V**	**9.00%**	**0.02V**	**0.80%**

Sysblocks


SIGNAL MANIPULATION
W1 Amp: 0
W2 Ofs: 0
BEGIN
matrix TSL
SYSBLOCKS
LINE IN
LINE OUT

CREATE HIGHLY FUNCTIONAL ELECTRONIC SYSTEMS USING PIC32 TECHNOLOGY TO MANIPULATE ANALOGUE SIGNALS USING FLOWCODE 10.1 AND SYSBLOCKS. FANTASTIC FOR DIGITAL AUDIO APPLICATIONS.

Flowcode Pro, PIC32 pack with Sysblocks board, cables and accessories bundle £610 + VAT
Flowcode with PIC32 pack with Sysblocks board, cables and accessories bundle £330 + VAT

10% off for PE readers using code: **PESYS10**

FLOWCODE 10.1 WITH MORE THAN 20 MICROCONTROLLER TARGETS INCLUDING ARDUINO UNO, PIC18877, RPI2B, AND ESP32 IS NOW FREE FOR HOBBYISTS.

☎ +44 (0) 1422 252 380 ✉ sales@matrixtsl.com

www.flowcode.co.uk

PART 2 OF JOHN CLARKE'S

SECURE REMOTE SWITCH

SECURE REMOTE SWITCH RECEIVER
Power
Relay
(Ack.)
Local
TIMER
ON
OFF
SECURE REMOTE SWITCH TRANSMITTER

This new Secure Remote Switch uses rolling codes for high security. The DC-powered receiver fits in a compact plastic case, so it can be mounted pretty much anywhere. After explaining how the circuitry works last month, this second and final article has all the construction details.

There are two versions of the keyfob transmitter; one uses a prebuilt transmitter module, while the other uses mostly discrete parts (with one extra IC) and is available as a complete kit.

Up to 16 transmitters can be used with one receiver, and multiple independent receivers can be built without the risk of the transmitters accidentally triggering the wrong receiver.

The receiver can be powered from 12V or 24V DC; there are slight component differences between the two options – the relay coil voltage varies, as does the value of one resistor. The receiver provides SPDT relay outputs that can switch low-voltage AC or DC up to 10A (possibly more if you choose a beefier relay).

Assuming you have gathered the parts, we will get straight into construction. After that will come the testing and setup instructions.

Performance

Both transmitter versions are built on PCBs measuring 29.8 × 39.4mm, with some common components including the SOIC-14 microcontroller, regulator, capacitors and a resistor. They vary in the UHF transmitter section, which can either be a prebuilt module (for the PCB coded 10109232) or built from discrete components (PCB coded 10109233).

The latter PCB includes more surface-mounting parts, making assembling slightly more challenging. However, it doesn't have any parts with particularly closely-spaced leads, so anyone with reasonable soldering skills should have a good chance of building it successfully.

Transmitter construction

The PCB overlays for the two transmitter boards are shown in Figs.4 & 5. Whichever transmitter you build, they are housed in a remote control enclosure that measures 37 × 63 × 17.5mm. This enclosure is designed for use with an A23 12V battery; you can also use an A27 12V battery with a smaller diameter but similar length.

The PCB is positioned within the enclosure by a moulded protrusion in the base that fits into a location hole in the PCB. This hole is just at the top edge of switch S2. Take care with the locating pin in the enclosure, as it can break easily.

If it is damaged, you can fix it by soldering a PCB pin into the locating pinhole on the PCB from the underside and drilling a 1mm hole into the keyfob base at the broken locating pin position. Trim the PCB pin at both ends so it's flush with the PCB on top and just long enough to meet flush with the underside of the enclosure when the PCB is installed.

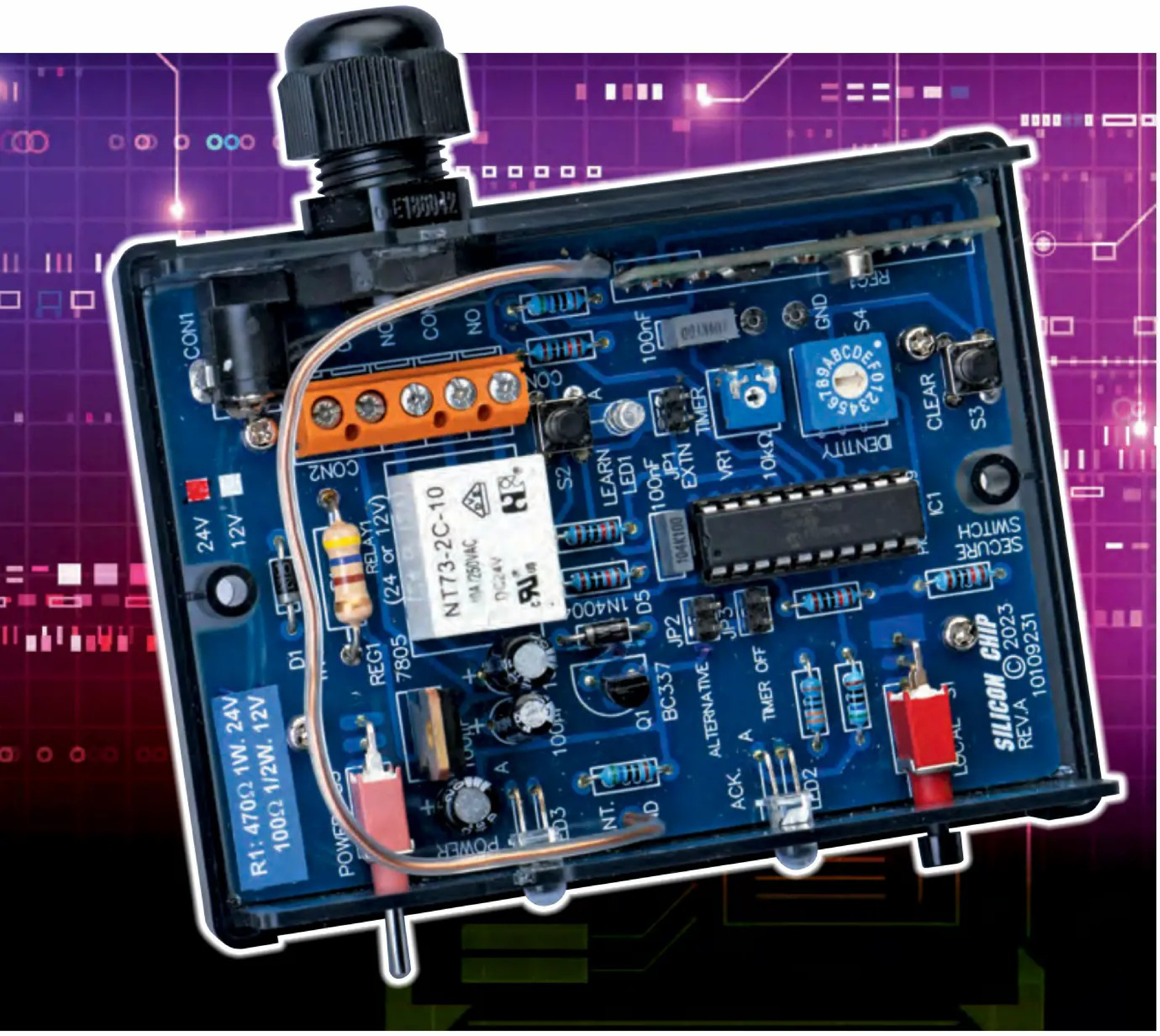
IC1 will need to be programmed before it is soldered. This IC can be obtained pre-programmed from Silicon Chip (individually or as part of a transmitter kit), or you can program it yourself if you have a suitable programmer. We described a programming adaptor that can be used for this type of chip in the September 2024 issue.

We'll start with the construction steps that apply to both versions, then follow with separate UHF transmitter assembly descriptions. The common parts are in the sections at the top and bottom of the transmitter PCB, with the parts that vary all being in the middle, below the row of switches and above the through-hole diode and SOT-223 package regulator.

Note that most SMD capacitors and inductors are unmarked, so you will need to rely on the packaging to show what they are and their value. Mount one component at a time to avoid mixing them up.

Start by fitting IC1, making sure it is orientated correctly. Solder pin 1 to the

Secure Remote Switch, part two



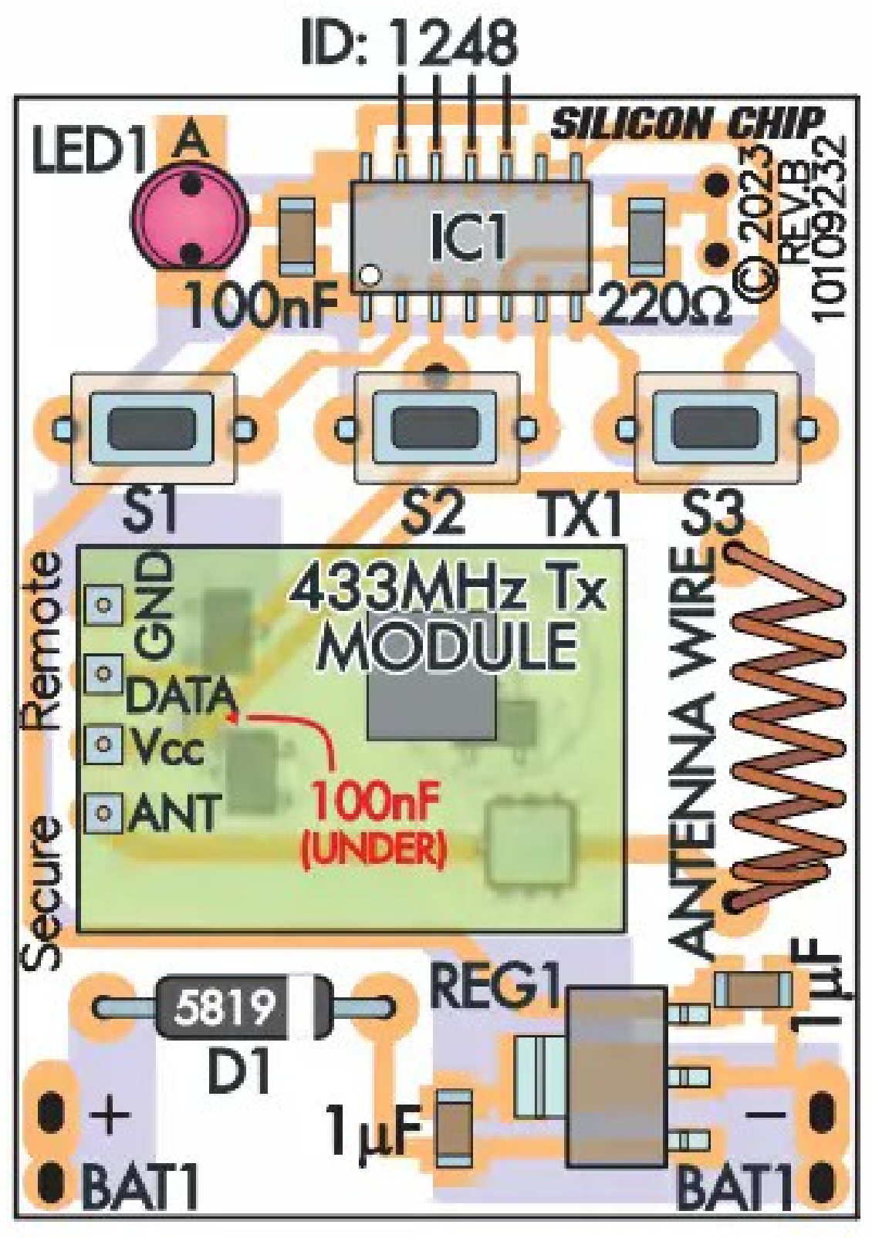


Fig.4 (top): bend the module leads so that the pins can be soldered as shown here, with GND at the top and ANT at the bottom. The battery clips are soldered to the pairs of slots in the two lower corners of the board.

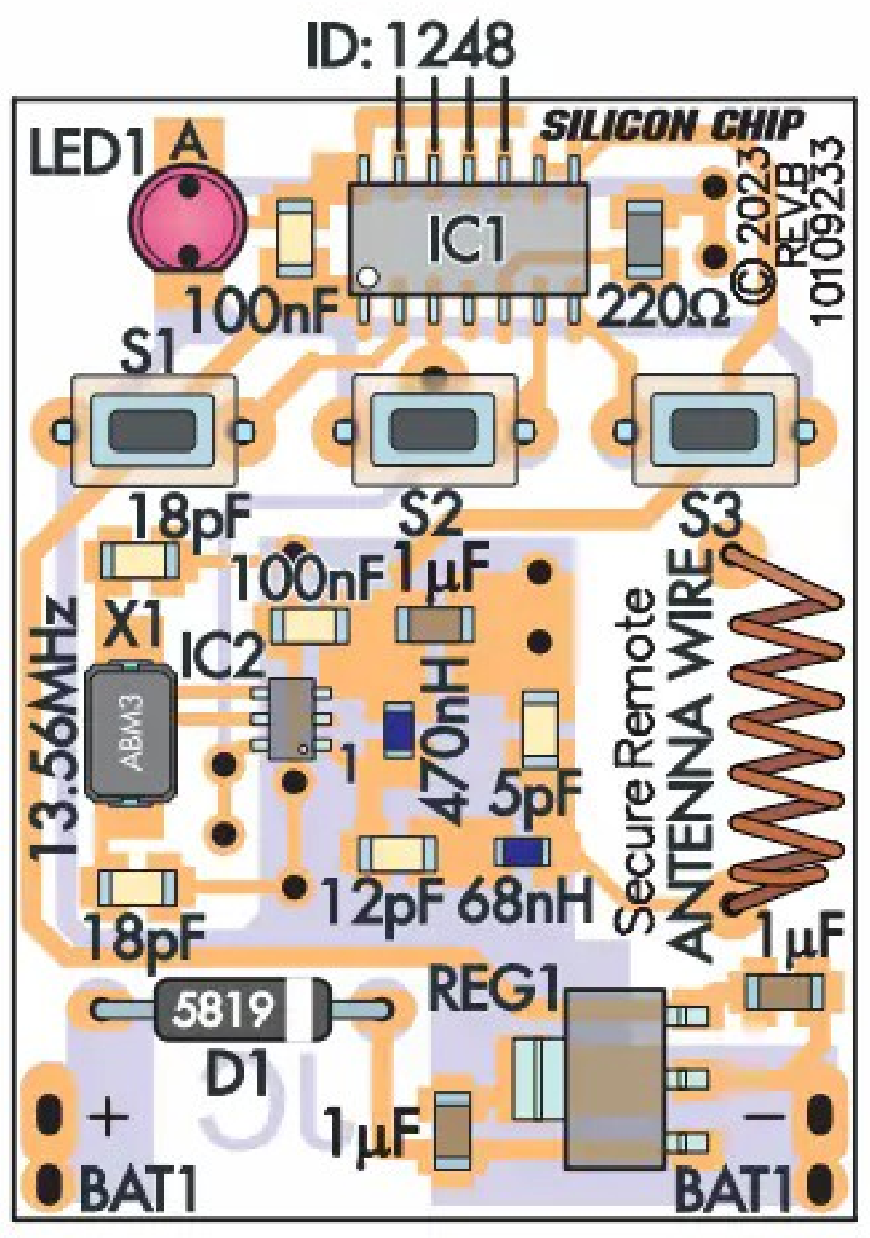


Fig.5 (bottom): on the discrete transmitter PCB, the only new polarised part is the transmitter IC (IC2). When soldering the two SMD inductors, you must position them so their exposed copper leads are in contact with the PCB.

PCB and check the alignment to ensure the IC pins all align with the PCB pads before soldering the remaining pins. Also check that it's sitting flat and not lifted on one side. After soldering, if any pins have a solder bridge between them, you can remove it with a dab of flux paste and some solder wick.

The Identity can be set at this stage. If only using one transmitter, it can be left at the default of '0' where none of the 1, 2, 4 or 8 links are made. For a different identity, connect one or more identity pins and the ground track using a solder bridge or a short wire soldered between the IC pins and the ground track. Table 5 (from last month) shows the 16 possible identity settings.

Next, fit the 220Ω resistor and 100nF capacitor at either end of IC1. To do this, tack solder one end of the component and remelt the solder to straighten it, if necessary, before soldering the opposite end. Then add a bit of fresh solder (or flux paste) to the first joint and heat it to re-flow it so it is nice and shiny.

Now install the three pushbutton switches, S1-S3. These are supplied with a kink in their leads and are more easily mounted if you straighten the leads first with pliers, then insert the switch leads into the allocated holes, pushing each switch down so its body is in contact with the PCB.

After that, install LED1, ensuring its polarity is correct (the longer lead is the anode [A]) and that the top of the LED lens is 10mm above the top surface of the PCB.

Mount REG1, diode D1 and the two 1µF capacitors next. D1 is a through-hole component that needs to be inserted into the PCB holes with the correct orientation. Solder REG1 in place by one pin first, then remelt that joint if necessary to align the pins correctly before soldering the remaining pins, then the tab. The two 1µF capacitors can be soldered similarly to the 100nF capacitor and 220Ω resistor.

The battery clips supplied with the enclosure should now be attached to the lower sides of the PCB. Solder these on both sides of the PCB, with the two prods inserted into the allocated slotted pads. Refer to our photos on page 75 to see how they should look once soldered in.

Our prototype isn't exactly the same as the final version, as we narrowed the prototype PCB slightly where the clips go. The final PCBs supplied will have a full-width PCB design that allows the clips to be captured in moulded L-shaped corrals in the base of the enclosure.

Module version parts

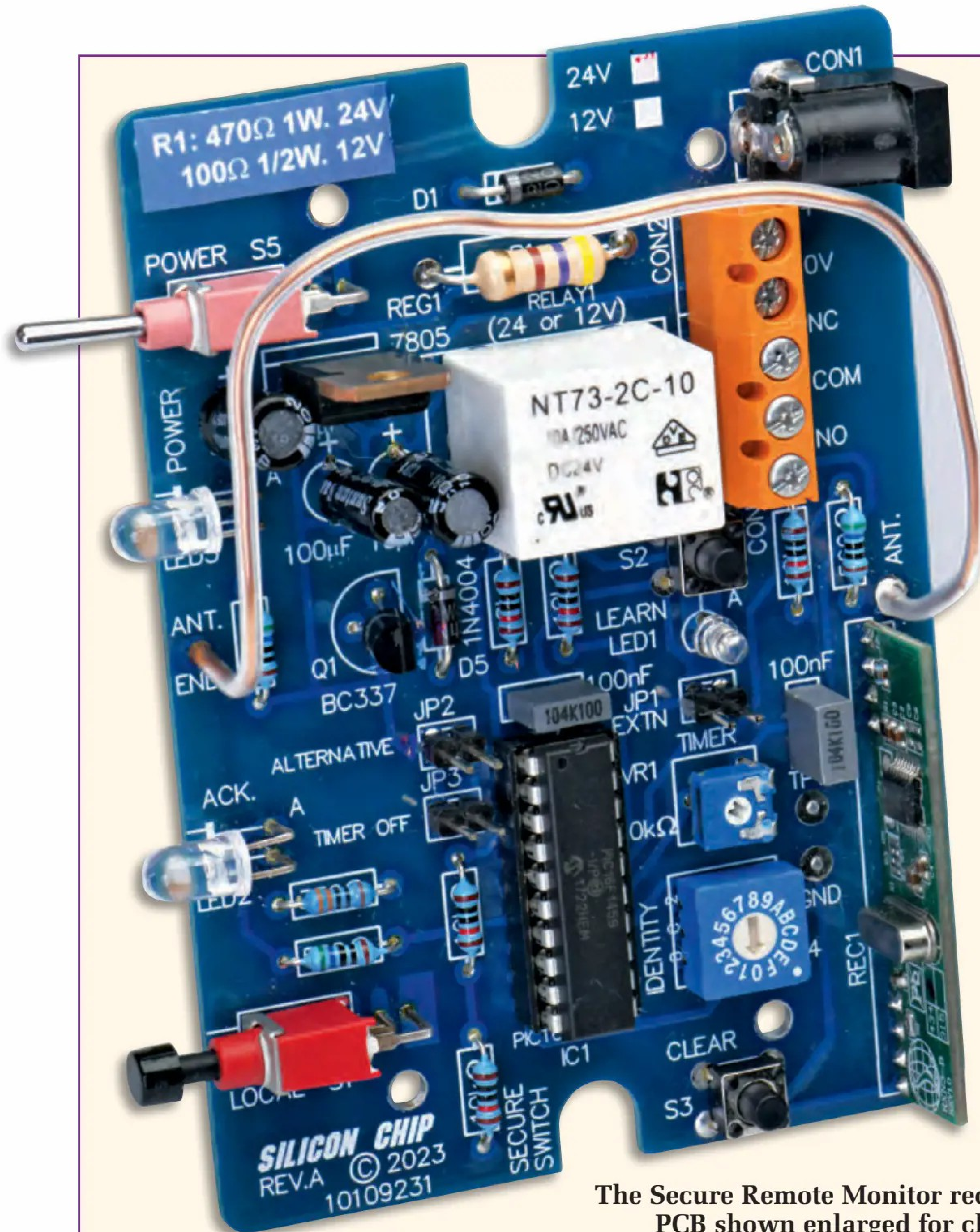
For the UHF module version (Fig.4), a 100nF capacitor needs to be soldered on the underside of the PCB; it is the only part on that side of the board. The UHF transmitter module can then be installed on the top side of the PCB, with its pins bent around the end of its PCB so it lies parallel to the main board, with a 1mm clearance between the main PCB. You can see how that was done on page 75.

The module's antenna is made from a 147mm length of 0.8mm diameter enamelled copper wire. Scrape 1mm of enamel off each end using a sharp craft knife, then close-wind seven turns on a 5.5mm diameter shaft (eg. the shank of a 5.5mm drill bit). Stretch the coil out to 13mm between the wire ends before soldering the ends to the PCB pads. The coil sits 1mm off the PCB.

Discrete version parts

Start with the discrete version parts by fitting IC2 – see Fig.5. Position it so the small pin 1 location dot aligns with that on the PCB. IC2 will have "F_113" etched on the top face. When it is orientated with the writing the right way up, pin 1 is at lower left.

Crystal X1 can be mounted next. It is soldered at the very ends of the component. We recommend you mount the capacitors before the two inductors


R1: 470Ω 1W. 24V
100Ω 1/2W. 12V
24V
12V
CON1
D1
POWER S5
REG1 7805
RELAY1 (24 or 12V)
CON2
0V
NC
COM
NO
NT73-2C-10
100µF
ANT.
END
Q1
BC337
1N4004
D5
LEARN LED1
S2
100nF
JP1 EXTN
JP2
JP3
ALTERNATIVE
ACK.
TIMER OFF
TIMER
VR1
IDENTITY
GND
REC1
IC1
CLEAR
S3
SECURE SWITCH
LOCAL S1
SILICON CHIP
© 2023
REV.A
10109231

The Secure Remote Monitor receiver PCB shown enlarged for clarity.

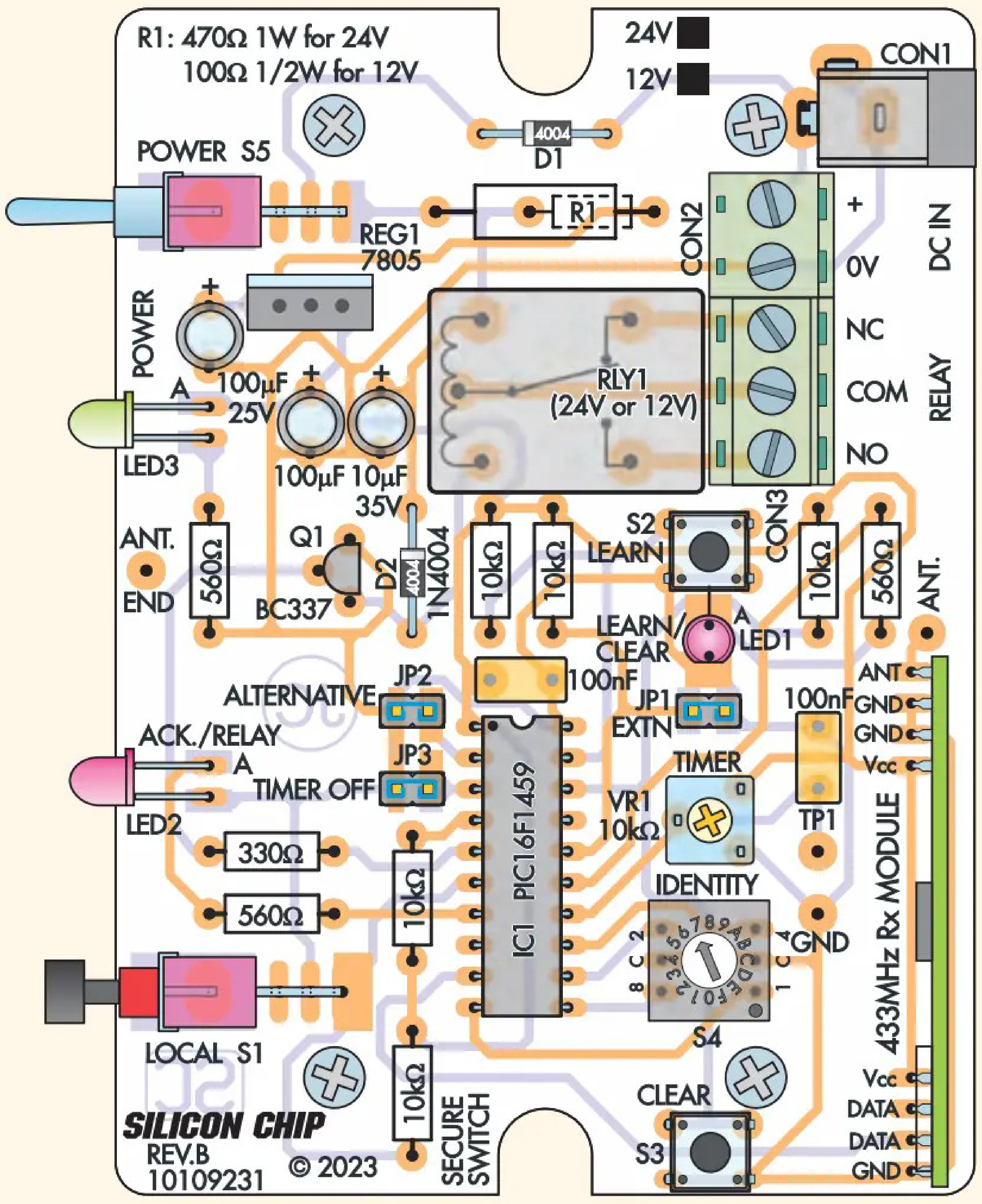

R1: 470Ω 1W for 24V
100Ω 1/2W for 12V
24V
12V
CON1
D1
4004
POWER S5
R1
REG1 7805
CON2
+
0V
DC IN
NC
COM
NO
RELAY
POWER
A
100µF 25V
100µF 35V
10µF
RLY1 (24V or 12V)
LED3
ANT.
END
560Ω
Q1
BC337
D2
1N4004
10kΩ
S2
LEARN
CON3
ANT.
LEARN/CLEAR
LED1
100nF
JP1
EXTN
ALTERNATIVE
JP2
ACK./RELAY
JP3
TIMER OFF
LED2
330Ω
560Ω
IC1 PIC16F1459
VR1 10kΩ
TIMER
TP1
IDENTITY
S4
GND
ANT
Vcc
433MHz Rx MODULE
DATA
LOCAL S1
CLEAR
S3
SECURE SWITCH
SILICON CHIP
REV.B
10109231
© 2023

Fig.6: the antenna wire is not shown on this diagram; refer to the photo above to see how it's routed between the two ANT pads on either side. The polarised components on this board are IC1, REG1, LED1-LED3, D1, D2, S4, the three electrolytic capacitors and the receiver module. Match the pin markings on the receiver module with those shown here.

(68nH and 470nH). Unlike the other passives, the inductors don't have pads on all four sides. Therefore, you must ensure their exposed leads are sitting on the PCB before soldering the ends. If you can't see this clearly, use a magnifying glass.

If you want to be sure that the components have been soldered correctly, trace the connections to the other sections of the PCB to where there should be continuity. Their inductance values are low enough that they will appear as short circuits (or at least low-resistance connections) to a multimeter.

For example, pin 3 of IC1 should provide a low resistance reading to pin 6 of IC2. Additionally, check that there are no short circuits between component pins on the PCB that shouldn't be connected.

The board assembly is completed by fitting the antenna. Make it from a 167mm length of 0.8mm diameter enamelled copper wire. Strip the insulation from each end by about 1mm using a sharp hobby knife and close-wind it on a 6.5mm shaft (eg, the shank of a 6.5mm drill bit). Stretch it out to 13mm end-to-end before soldering in with a 1mm coil clearance above the PCB.

Receiver construction

The 70 × 96.5mm receiver PCB is coded 10109231 – see Fig.6. All the onboard components are through-hole types, giving you a break from the surface-mounting parts that were on the transmitter. The assembled PCB fits nicely in a Ritec enclosure that measures 105 × 80 × 33mm.

Install the resistors first, taking care to place each in its correct position. The resistor colour codes were shown in the parts list last month, but you should also use a digital multimeter to check each resistor before mounting it in place. Note the different R1 value for 24V use (470Ω 1W) compared to 12V (100Ω ½W or 1W).

Diodes D1 & D2 are next. Make sure these are orientated correctly before soldering their leads. Then install the socket for IC1, ensuring its notched end matches the position shown in Fig.6. Wait to fit IC1 as that step comes later, after the power supply has been checked.

Regulator REG1 is mounted vertically on the PCB as far down as it will go, to allow clearance for the lid when in the enclosure.

Secure Remote Switch, part two

Next, install trimpot VR1, transistor Q1 and the BCD switch (S4). S4 must also be orientated as shown. Switches S2 and S3 can also be mounted now.

The capacitors can then be fitted. The electrolytic capacitors are polarised and must be installed with the polarity shown (the longer lead is positive). Pay attention to the voltage ratings for the 10µF and the 100µF capacitors if you intend to use a 24V supply. For a 12V supply, 16V-rated capacitors can be used throughout. You can install the two 100nF MKT polyester capacitors either way around.

LED1 mounts with the top of the lens up to 12mm above the surface of the PCB and the anode (longer lead) to the hole marked “A”.

Switches S1 and S5 can be installed now, taking care to use the toggle switch at the S5 location and the pushbutton switch for S1. The two remaining LEDs (LED2 and LED3) mount horizontally with leads bent at right angles 6mm back from the rear of the package. Make sure you bend the leads so the longer anode lead is in the “A” pad. The height of the LED centres should be 5mm above the PCB’s top face.

CON1 is the PCB-mounting barrel socket, while CON2 and CON3 are 2-way and 3-way screw terminals. Dovetail CON2 and CON3 together by sliding them along the side mouldings to produce a 5-way connector. Orientate all these connectors so the openings are toward the rear of the PCB, then solder them in place.

Mount relay RLY1 next. Ensure you use a 24V coil relay if you will use a 24V DC supply or a 12V coil relay for 12V use.

Now fit the headers for jumpers JP1, JP2 and JP3 and install the 433.9MHz receiver module. Before soldering the receiver module, compare the pin labels on the back of the module to those in Fig.6 to ensure it is the right way around; there are two possible ways it could be fitted, and only one is correct. Your module might differ from ours, so don’t rely on the photos; check the pin connections.

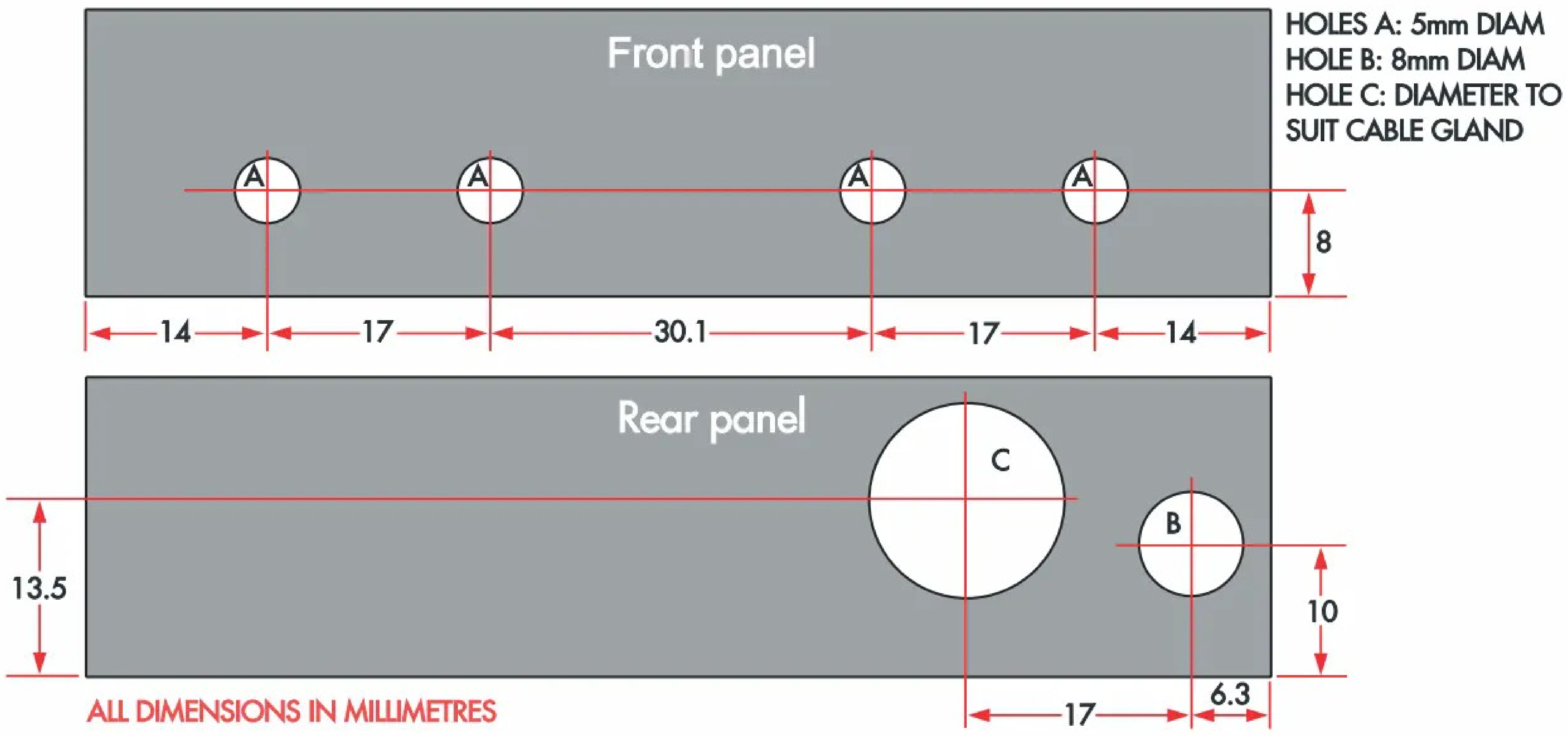


Fig.7: the front and rear panel drilling details. The large hole marked “C” on the rear panel is for a cable gland that secures the wires to the relay terminals.

The antenna (not shown in Fig.6) is made from a 169mm length of 0.8mm diameter enamelled copper wire. It extends from the antenna pad adjacent to the UHF receiver to another pad between LED2 and LED3.

We covered it with 1mm heat shrink tubing. That is not really required, but it produces smoother bends in the wire as the antenna is shaped. Make sure to scrape away the enamel insulation from both ends of the antenna wire before soldering it into position.

Testing

IC1 will need to be programmed before use. You can order a preprogrammed device from Silicon Chip (either individually or as part of a short-form receiver kit). You can also program it yourself using the hex file available from our website.

Before plugging in IC1, apply power and check that the voltage between pins 1 and 20 of its socket measures close to 5V (4.75-5.25V). If this is correct, switch the power off and insert IC1 into the socket, taking care to orientate it correctly (with its pin 1 end at the notched end of the socket).

Case preparation

The front and rear panels need holes drilled to allow the LEDs and switches to protrude through and for access to the relay contact screw terminals and DC socket at the rear. Fig.7 shows all the panel drilling details.

There is provision for a cable gland to secure any wires connecting to the screw terminals. Either a PG7 or PG9-sized gland will fit. When using a PG9 gland, the nut that secures the gland to the back of the panel will need to have the circular fused-on washer cut back to be flush with the straight sides of the nut.

To do that, only the washer sections on directly opposite sides of the nut need to be brought back to the shape of the hexagonal nut so those sides of the nut can sit flush on the PCB and top lid of the enclosure. This can be done with side cutters and a file.

The panel artwork (Fig.8) can be downloaded from our website as a PDF file and printed onto a sticky-backed label. We have instructions on making labels at pemag.au/Help/FrontPanels

Once made, the labels can be affixed to the panels after drilling. Cut out the holes in the label with a sharp craft knife. There is also artwork to make labels for the transmitters. The

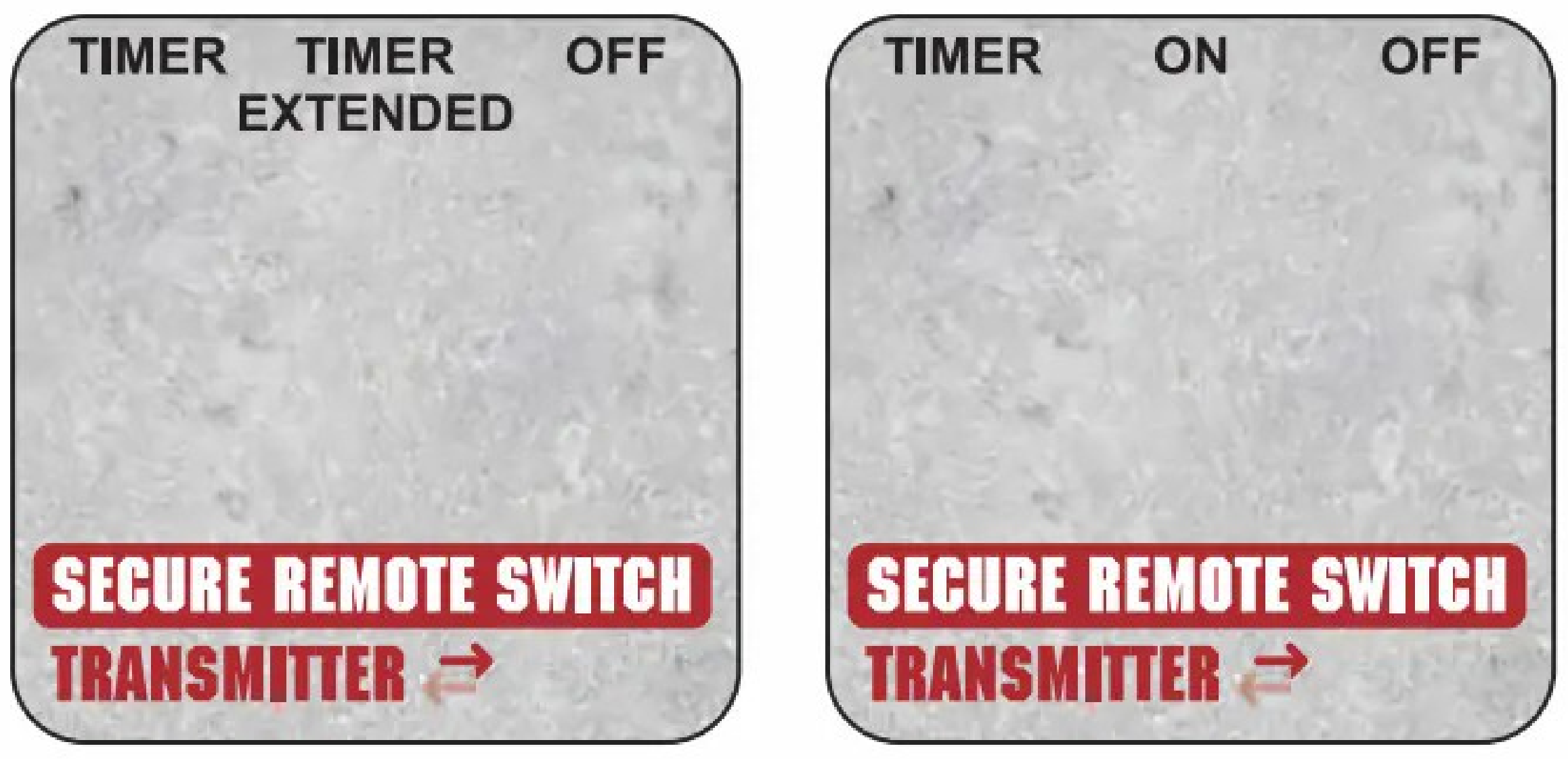
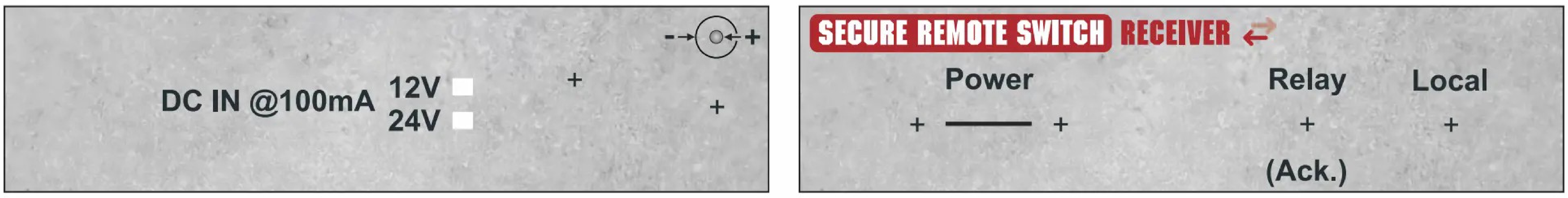


Fig.8: you can download this panel label artwork from the Silicon Chip website, print it onto adhesive stock and stick it to the front and rear case panels. Stickers are also supplied with the transmitter kits.



Rolling Code Systems – frequently asked questions

One question that's often asked about rolling code systems is what happens if one of the switches on the transmitter is pressed when the transmitter is out of range of the receiver. Will the receiver still work when the transmitter is later brought within range, and the button pressed again?

This question is asked because the code the receiver was expecting has already been sent (but not received), and the transmitter has rolled over to a new code. How does the system get around this problem?

The answer is that the receiver will process a signal that is the correct length and data rate, but it will not trigger the relay unless it receives the correct code. So if the signal format is valid, but the code is incorrect, the receiver then calculates the next code that it would expect and checks this against the received code. If it matches, the receiver will trigger the relay; that means it missed one button press.

If the code is still incorrect, the receiver calculates the next expected code, and it will do this up to 10 times, to handle cases where there have been multiple transmitter button presses out of range.

If none of these are correct, the receiver then changes its operation to allow for a possible valid signal more than 10 codes ahead. The receiver waits for two valid separate transmission codes before restoring correct operation.

On the first receipt of a valid transmission, it looks ahead up to 200 codes, so it is more likely the required valid code will be found, but it doesn't trigger the relay. The Learn LED lights during this look-ahead operation. If a valid code is found, the receiver waits for the next code sent by the transmitter. This following code must also be correct before the receiver will operate the relay.

If only one or neither code is correct, the receiver will not take action. If it's more than 200 codes ahead, the transmitter will need to be re-registered to operate the receiver.

You can test this process by switching the receiver off and pressing one of the remote control switches 10 times or more. Then switch on the receiver and press one of the switches again. The Learn LED will light, indicating that the look-ahead feature beyond the initial 10 times is activated. The selected function on the remote should operate on the next press of the switch, and the Learn LED extinguishes.

While two successive transmission codes could be intercepted, recorded and re-sent in an attempt to activate the receiver, these codes will not be accepted by the receiver. That's because they have presumably already been received and processed, and the receiver has already rolled past those codes. It will look forwards but not backwards.

Another transmitter with a different identity will still operate the receiver (provided it has been synchronised in the first place). That's because the receiver tracks each transmitter's rolling codes separately.

Calculating the code

Another question that's often asked is how the receiver knows which code to expect from the transmitter since it changes each time. The answer is that the transmitter and the receiver both use the same series of calculations to determine the next code. They also both use the same variables in the calculation; unique values that no other transmitter uses.

For our Secure Remote Switch, we use a linear congruential generator (LCG) in conjunction with a 31-bit pseudo-random number generator (PRNG).

The LCG uses an initial seed value, an addition value and a multiplication factor to produce a nominally random result.

For example, if consecutive codes have the number 3 added and then multiplied by 49, with the same starting number, both the transmitter and receiver will go through the same sequence. But the actual numbers used are very large, making it difficult to predict the next code by peeking at a few values in the sequence.

The code is 48 bits long, with 281,474,976,710,656 possible values (that's over 281 quintillion or 2.8 x 10^{14}).

One problem with the LCG is that it can produce recurring values; depending on the factors, it can produce the same number more than once within a few hundred rolling code calculations. To prevent this, we include a second layer of randomisation. So once we have the value from the LCG calculation, this value is used in the PRNG to generate a pseudo-random number for the rolling code.

The PRNG randomisation runs between one and 256 times before providing the 'random' number for the rolling code value. The number generated is then used as the seed in the LCG for generating the next number in the sequence. Using the PRNG makes it difficult to predict the rolling code sequence even if the multiplier or addition value for the LCG is known.

We throw further complications by also using code scrambling. The calculated code is not sent in the same sequence each time. There are 32 possible scrambling variations that are applied to the code, so predicting the next code becomes very difficult.

What if the transmitter rolling code is identical for two consecutive codes, and the first of these identical codes is intercepted and re-transmitted to open the lock? Our system has safeguards to prevent the same code from appearing twice in succession. There is a check for the same code repeating itself for consecutive codes. If the code is the same, the duplicate is effectively skipped, preventing this possibility.

Multiple transmitters

Wouldn't the receiver lose its synchronisation if several transmitters were used? No, because each transmitter operates independently. Each of the 16 possible transmitters used with a given receiver has its own different identity from one to 16.

The codes sent by each transmitter are different due to the unique identifier within each transmitter IC that sets the rolling code calculation. Also, the code includes the transmitter identity value that differs between each transmitter. The receiver stores up to 16 different rolling code and calculation parameters, one for each identity, so each transmitter is treated independently.

Therefore, even if one transmitter is not used for months while other transmitters are used frequently, its rolling codes will remain synchronised with the receiver.

Secure Remote Switch, part two

On the transmitter, S1 is red, S2 is blue and S3 is black.

SECURE REMOTE SWITCH RECEIVER
Power
Relay
(Ack.)
Local

two variations cater for the timer options, as shown in Table 2 last month, set using JP2.

Note that the rear panel artwork and the receiver PCB have square white blocks to allow you to mark the power supply voltage required. Use a marker pen to colour the squares depending on whether the board has been built for a 12V or 24V supply.

Four self-tapping screws are provided with the receiver enclosure to secure the PCB to the base. Similarly, two screws are supplied to secure the two halves of the enclosure.

Registering a transmitter

To register the transmitter with the receiver, press the Learn switch (S2) on the receiver. The Learn/Clear LED (LED1) will light.

On the transmitter, remove the battery and reinsert it while pressing and holding switch S1. This will set the transmitter to Synchronisation mode (with its Acknowledge LED lit) and send the registering code when S1 on the transmitter is released and then pressed again.

The rolling code is continuously updated during the synchronisation time between when S1 is released and when it is pressed again. This randomises the rolling code generation sequence to an undetermined point, due to the rapid rate of the code recalculation. On average, it is updated around 500 times per second. The rolling code is then well into its generating sequence.

This makes it hard to guess the code based on possible MUI values, even if the initial seed value for the code generation is known.

The acknowledge LED on the receiver will flash twice, and the Learn LED will extinguish once registration is complete. If it does not seem to work, try this registration procedure again. Test the transmitter and check that the receiver responds by switching the relay on and off. It will take a couple of attempts before the transmitter and receiver start working together.

Deregistering a lost transmitter

Any transmitter that has been registered can be prevented from operating the receiver; for example, if a transmitter is lost and you don't want it to be used by an unauthorised person.

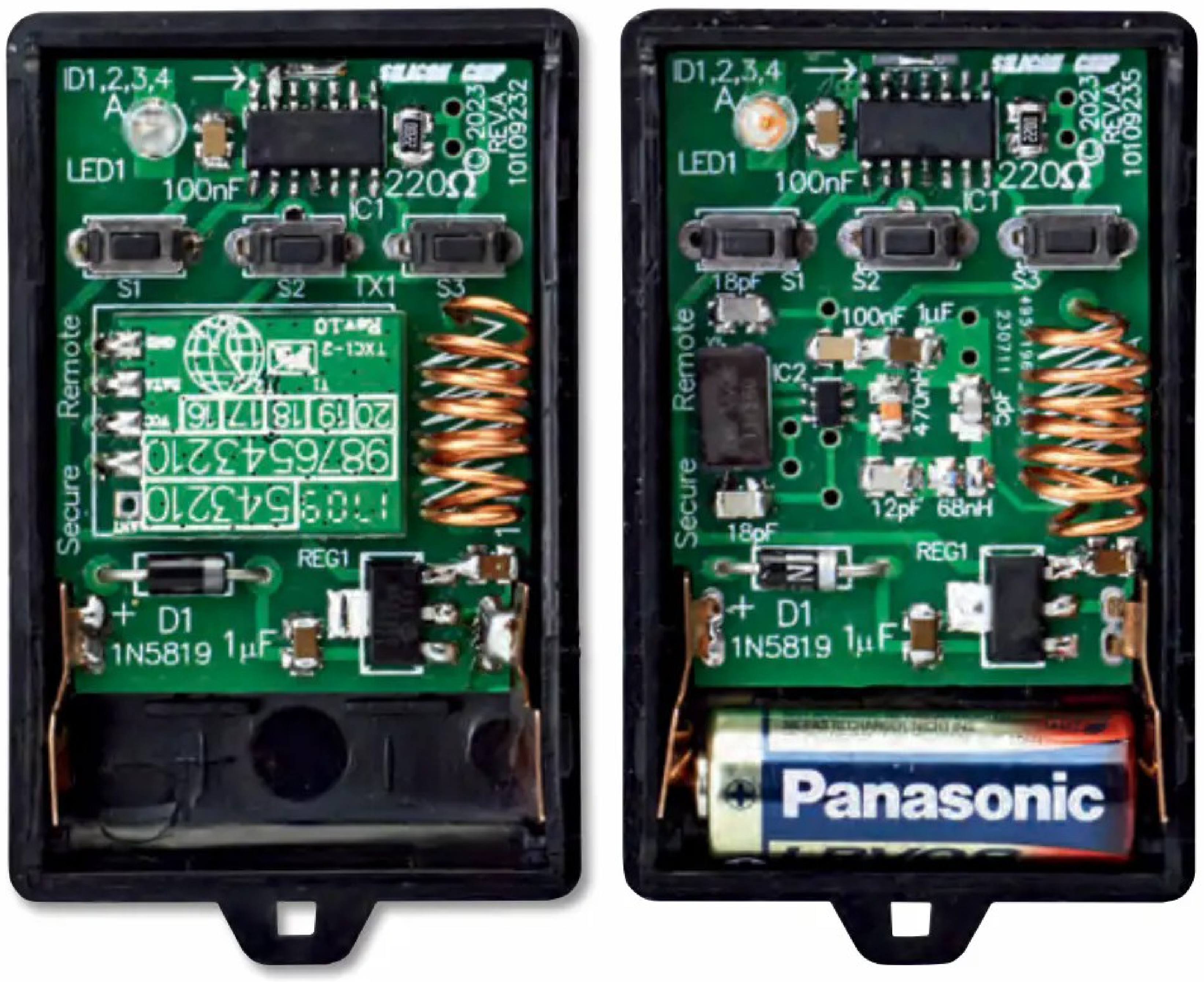
Deregister the lost transmitter by selecting the transmitter's Identity using BCD switch S4. The switch is labelled 0 to F, where the labels A-F correspond to identities 10-15. Then press and hold the Clear button (S3) for over one second. The Learn/Clear LED will light initially, then extinguish after S3 is released and the transmitter is deregistered.

If you are unsure of the Identity of the lost transmitter, you can deregister all the registered transmitters, one at a time, then re-register the other transmitters again.

Jumper options

There are three jumper positions on the receiver board; Table 1 to Table 4, published last month, show what they do.

JP1 selects the timer length multiplier (see Table 1). The range is 250ms to 60s with JP1 out (the ×1 range) or 60s to 4.5 hours with JP1 in (the ×255 range). Table 4 shows typical timeouts versus TP1 voltages as trimpot VR1 is adjusted. JP2 affects the function of the buttons on the remote control, as shown in Table 2. JP3 affects the function of switch S1 on the receiver, as shown in Table 3.



The module-based (left) and discrete (right) versions of the transmitter PCB shown enlarged for clarity. We have used an A23 12V battery, which fits snugly with the recommended battery clips.

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
JANUARY 2025		
USB to PS/2 Converter for VGA PicoMite	07111231	£3.99
ps2x2pico USB to PS/2 Converter	07111232	£3.99
Compact Microphone Preamplifier (case version)	01110231	£5.99
Compact Microphone Preamplifier (integrated version)	01110232	£5.99
DECEMBER 2024		
Raspberry Pi Clock Radio PCB pair	19101241/2	£12.99
VGA PicoMite (blue)	07107221	£5.99
Secure Remote Switch receiver	10109231	£4.99
Secure Remote Switch transmitter (module version)	10109232	£2.99
Secure Remote Switch transmitter (discrete version)	10109233	£2.99
NOVEMBER 2024		
Ideal Diode Bridge Rectifiers (28mm square)	18101241	£3.99
Ideal Diode Bridge Rectifiers (22mm square)	18101242	£3.99
Ideal Diode Bridge Rectifiers (4-pin SIL)	18101243	£3.99
Ideal Diode Bridge Rectifiers (tiny SOT-23)	18101244	£3.99
Ideal Diode Bridge Rectifiers (standalone SMD)	18101245	£4.99
Ideal Diode Bridge Rectifiers (standalone TH; 2oz copper)	18101246	£6.99
Multi-channel Volume Control (volume PCB)	01111221	£4.99
Multi-channel Volume Control (control PCB)	01111222	£4.99
Multi-channel Volume Control (OLED PCB, black)	01111223	£4.99
OCTOBER 2024		
TQFP-32 Programming Adaptor	24108231	£4.99
TQFP-44 Programming Adaptor	24108232	£4.99
TQFP-48 Programming Adaptor	24108233	£4.99
TQFP-64 Programming Adaptor	24108234	£4.99
2m VHF FM Test Signal Generator	06107231	£4.99
SEPTEMBER 2024		
Calibrated Microphone (SMD version)	01108231	£3.99
Calibrated Microphone (TH version)	01108232	£3.99
PIC Programming Adaptor	24105231	£4.99
Salad Bowl Speaker crossover	01109231	£7.99
AUGUST 2024		
Arduino ESR Meter combined L/C-ESR	04106182	£9.99
Arduino ESR Meter separate ESR	04106183	£7.99
Watering System Controller	15110231	£11.99
JULY 2024		
Dynamic NFC Tag (22 × 31mm)	06101231	£5.95
Dynamic NFC Tag (86 × 54mm)	06101232	£7.95
Reciprocal Frequency Counter	CSE230101C	£8.95
Reciprocal Frequency Counter front panel (black mask)	CSE230102	£7.95
Pi Pico-based Thermal Camera	04105231	£6.95
JUNE 2024		
Loudspeaker Test Jig	04106231	£12.95
MAY 2024		
GPS-Disciplined Oscillator	04103231	£12.95
Dual RF Amplifier	CSE220602A	£5.95
Songbird	08103231	£7.95
APRIL 2024		
Basic RF Signal Generator	CSE221001	£8.99
Basic RF Signal Generator front panel (black solder mask)	CSE220902B	£9.99
Silicon Chirp Cricket	08101231	£7.99
Test Bench 'Swiss Army Knife'	04110221	£9.99

PROJECT	CODE	PRICE
MARCH 2024		
Digital Volume Control Potentiometer SMD version	01101231	£8.95
Digital Volume Control Potentiometer through-hole version	01101232	£9.95
Advanced SMD Test Tweezers body	04106221	£6.95
Advanced SMD Test Tweezers legs (pair)	04106212	£7.95
FEBRUARY 2024		
Active Mains Soft Starter	10110221	£10.95
JANUARY 2024		
Q Meter	CSE220701	£8.95
Q Meter front panel (black solder mask)	CSE220704	£7.95
Raspberry Pi Pico W BackPack	07101221	£8.95
DECEMBER 2023		
Digital Boost Regulator	24110224	£8.95
Dual-Channel Power Supply for Breadboards	04112221	£11.95
Display Adaptor for the Breadboard PSU	04112222	£9.95
NOVEMBER 2023		
LC Meter Mk3	CSE220503C	£9.95
DC Supply Filter for vehicles	08108221	£8.95
Discrete Audio Op Amp PCB	AO1-JUL23	£9.95
Discrete Audio Op Amp PCB + essential components	AO2-JUL23	£17.95
OCTOBER 2023		
Buck/Boost Charger Adaptor	14108221	£9.95
PIC Breakout Board for SOIC parts	24110225	£5.95
PIC Breakout Board for DIP parts	24110222	£5.95
AVR64DD32 Breakout board	24110223	£5.95
Automatic Train Controller	09109221	£6.95
Chuff Sound module	09109222	£6.95
SEPTEMBER 2023		
Mini LED Driver	16106221	£7.95
New GPS-Synchronised Clock	19109221	£9.95
AUGUST 2023		
110dB RF Attenuator	CSE211003	£9.95
Universal Battery Charge Controller (2023 update)	14107192	£9.95
Wide-Range OhmMeter	04109221	£12.95
JULY 2023		
Multimeter Checker/Calibrator	04107221	£10.95
JUNE 2023		
Arduino Programmable Load	04105221	£8.95
Buck-Boost LED Driver	16103221	£8.95
MAY 2023		
Precision AM-FM DDS Signal Generator	CSE211002	£14.95
Improved SMD Test Tweezers programmed PIC	0410621PIC	£12.95
AO 2x-dual-to-quad through-hole (pack of five)	AO1-MAY23	£6.95
AO 2x-dual-to-quad SMD (pack of five)	AO2-MAY23	£6.95
AO 2x-single-to-dual through-hole (pack of five)	AO3-MAY23	£6.95
AO 2x-single-to-dual SMD (pack of five)	AO4-MAY23	£6.95
APRIL 2023		
500W Amplifier Module	see p.22, April 2023	
Clipping Indicator (per channel)	01112211	£7.95
CD Welder Power Supply (one needed)	29103221	£9.95
CD Welder Controller (one needed)	29103222	£9.95
CD Welder Energy Storage module (several needed)	29103223	£7.95
AO Universal Dual Op Amp Board	AO1-APR23	£9.95
AO Stereo RIAA precision passives kit for dual op amp	AO2-APR23	£8.95

PCBs for most recent *PE/EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
MARCH 2023		
Pico BackPack	07101221	£9.95
Semaphore Signal (controller)	09103221	£7.95
Semaphore Signal (blade)	09103222	£5.95
FEBRUARY 2023		
Tesla Coil driver board	26102221	£9.95
Tesla Coil potentiometer board	26102222	£5.95
Cooling Fan Controller & Loudspeaker Protector	01102221	£8.95
Remote Gate Controller	11009121	£12.95
JANUARY 2023		
Classic LED Metronome – 8-LED	23111211	£7.95
Classic LED Metronome – 10-LED	23111212	£8.95
Multi-Channel Speaker Protector – 6 channel	01101221	£9.95
Multi-Channel Speaker Protector – 4 channel	01101222	£7.95
Remote Control Range Extender – IR-to-UHF	15109212	£5.95
Remote Control Range Extender – UHF-to-IR	15109211	£7.95
AO Universal Single Op Amp Board	AO1-JAN23	£7.95
DECEMBER 2022		
Hummingbird Amplifier	01111211	£9.95
SMD Trainer PCB	29106211A	£8.95
SMD Trainer PCB + parts	29106211B	£13.95
NOVEMBER 2022		
USB Cable tester – main PCB	04108211	£12.95
USB Cable tester – front panel	04108212	£5.95
USB Cable tester – optional panel	SC5970	£5.95
Model Railway Carriage Lights – PCB	09109211	£6.95
AO trnsfmr PCB – standard VTX-A range	VTX-101-007	£6.95
AO trnsfmr PCB – dual-outline VTX102-3001/101-3002	VTX-Dual	£6.95
OCTOBER 2022		
SMD Test Tweezers – PCB and pair of tweezer arms	04106211-2	£11.95
SMD Test Tweezers – programmed PIC12F1572-I/SN	0410621A	£7.95
Tele-com	12110211	£12.95
SEPTEMBER 2022		
Touchscreen Digital Preamp – main board	01103191	£12.95
Touchscreen Digital Preamp – adaptor board pair	01103192	£5.95
20A DC Motor Speed Controller	11006211	£9.95
AUGUST 2022		
Multi-purpose Battery Manager – I/O Expander	11104212	£5.95
Multi-purpose Battery Manager – Switch Module	11104211	£8.95
Simple MIDI Music Keyboard (for 8 switches)	23101213	£6.95
Nano Pong	08105212	£7.95
JULY 2022		
Silicon Labs AM/FM/SW Radio	CSE210301C	£10.95
Level Crossing Controller	09108211	£6.95
JUNE 2022		
Full-wave Motor Speed Controller	1010221	£8.95
PIC Programming Helper for 8-pin PICs only	24106211	£7.95
PIC Programming Helper for 8, 14 or 20-pin PICs	24106212	£10.95
Advanced GPS Computer	05102211	£9.95
MAY 2022		
Bus board PCB for Analogue Vocoder	AO1-MAY22	£10.95
Complete set of 14 PCBs for Analogue Vocoder	AO2-MAY22	£97.95
Programmed EEPROM for Digital FX Unit	FX1-MAY22	£10.95
Programmed PIC for Digital FX Unit w/ potentiometer	FX2-MAY22	£8.95

PROJECT	CODE	PRICE
APRIL 2022		
64-key MIDI Matrix shield	23101211	£8.95
64-key MIDI Matrix switch board	23101212	£11.95
High-current Battery Balancer	14102211	£10.95
Digital FX Unit – using potentiometer	01102211	£9.95
Digital FX Unit – using BCD switch	01102212	£9.95
Universal Audio PSU	AO1-APR22	£11.95
MARCH 2022		
Mini Isolated Serial Link	24102211	£5.95
Busy Loo Indicator	16112201	£5.95
Analogue Vocoder – Band-pass filter board	AO1-MAR22	£9.95
Analogue Vocoder – HP/LP filter board	AO2-MAR22	£9.95
FEBRUARY 2022		
Arduino-based Power Supply	18106201	£9.95
Battery Monitor Logger 11106201	£10.95	
Electronic Wind Chimes 23011201	£10.95	
Analogue Vocoder – Driver Amplifier	AO-FEB22	£8.95
JANUARY 2022		
Vintage battery Radio Li-ion Power Supply	11111201	£9.95
MiniHeart: A Miniature Heartbeat Simulator	01109201	£8.95

For the many earlier PCBs that we stock please see the PE website: www.electronpublishing.com

PE/EPE PCB SERVICE

Order Code   Project   Quantity   Price

....

....

....

....

....

....

....

Name ....

Address ....

....

Tel ....

Email ....

Card No ....

Valid From ....Expiry Date ....

Card Security No ....

You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com

No need to cut your issue – a copy of this form is just as good!

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

Practical Electronics, Electron Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 01202 880299 Email: pe@pemag.au
On-line Shop: www.electronpublishing.com

Sorry, cheques cannot be accepted as we are based in Australia and our UK bank account does not allow cheque deposits.

NOTE: Most boards are in stock and sent within seven days of order receipt. Please allow up to 28 days delivery if we need to restock.

Practical Electronics PCB SERVICE

PROJECT	CODE	PRICE
DECEMBER 2021		
AM/FM/SW Digital Receiver	CSE200902A	£13.95
Balanced Input and Attenuator for USB CODEC	01106202	£11.95
NOVEMBER 2021		
Dual Battery Lifesaver	11111202	£6.95
OCTOBER 2021		
Mini Wi-Fi LCD BackPack	24106201	£8.95
SEPTEMBER 2021		
USB SuperCodec PCB	01106201	£14.95
Audio DDS Oscillator PCB	01110201	£5.95
Audio DDS Oscillator rotary encoder	01110201-ENC	6.95
Programming Adaptor Board for Audio DDS Oscillator	01110202	£5.95
High-power Ultrasonic Cleaner main PCB	04105201	£14.95
High-power Ultrasonic Cleaner front-panel PCB	04105202	
Night Keeper Lighthouse PCB	08110201	£5.95
AUGUST 2021		
Ol’ Timer PCB	19104201	£11.95
Ol’ Timer 8x8 RGB LED module using WS2812B	19104201-88	£8.95
Ol’ Timer set of acrylic case pieces and spacer	19104201-ACR	£8.75
Ol’ Timer DS3231 RTC IC wide SOIC-16	19104201-RTC	£5.95
Wideband Digital RF Power Meter	04106201	£9.75
Switchmode 78xx regulators ***(PACK of 5!)***	18105201	£7.95
Cool Beans SMAD display	CB-AUG21	£11.95
JULY 2021		
ATtiny816 Breakout / Dev Board w/ Capacitive Touch	24110181	£9.75
IR Remote Control Assistant (Jaycar version)	15005201	£8.95
IR Remote Control Assistant (Altronics version)	15005202	£8.95
PIC18F Development Board	PNM-JUL21	£12.95
Microphone Preamplifier	AO-JUL21	£11.95
JUNE 2021		
Roadies’ Test Signal Generator (surface-mount ver.)	01005201	£8.95
Roadies’ Test Signal Generator (through-hole version)	01005202	£9.95
Touchscreen Wide-range RCL Box (Resistor module)	04104201	£18.95
Touchscreen Wide-range RCL Box (Ind/Cap module)	04104202	
KickStart Part 3 – Gyrator-based Audio Filter	KS3-2021	£7.95
MAY 2021		
7-Band Equaliser (Mono)	01104201	£8.95
7-Band Equaliser (Stereo)	01104202	£10.95
Car Altimeter	05105201	£7.95
APRIL 2021		
Reflow Oven – DSP Active Crossover (CPU)	01106193	£19.95
Reflow Oven – DSP Active Crossover (Front panel)	01106195	
Reflow Oven – DSP Active Crossover (LCD)	01106196	
Frequency Reference Signal Distributor	CSE200103	£8.95
MARCH 2021		
Nutube Guitar Effects Pedal	01102201	£12.95
Programmable Thermal Regulator (Peltier Interface)	21109181	£18.95
Programmable Thermal Regulator (Peltier Driver)	21109182	
Tunable HF Preamp	CSE190502	£8.95
FEBRUARY 2021		
4G Remote Monitoring	27111191	£9.95
JANUARY 2021		
Nutube Valve Preamplifier	01112191	£12.95
Arduino DCC Controller	09207181	£10.95

PROJECT	CODE	PRICE
DECEMBER 2020		
Pseudo-Random Sequence Generator	16106191	£7.95
Clever Charger	14107191	£11.95
PE Theremin Amplifier	AO-1220-01	£8.95
NOVEMBER 2020		
LED Christmas Tree (1 off)	16107181-1	£6.95
LED Christmas Tree (4 off)	16107181-2	£14.95
LED Christmas Tree (12 off)	16107181-3	£24.95
LED Christmas Tree (20 off)	16107181-4	£34.95
USB/SPI Interface Board	16107182	£5.95
45V/8A Power Supply PCB *plus* acrylic spacer	18111181	£14.95
45V/8A Power Supply five-way display bezel	18111181-BZ	£3.95
Five-way LCD Panel Meter/Display	18111182	£7.95
OCTOBER 2020		
Digital Audio Millivoltmeter	04108191	£9.95
Precision Signal Amplifier	04107191	£7.95
SEPTEMBER 2020		
PE Theremin PSU	AO-0920-01	£5.95
PE Theremin PSU transformer	AO-0920-02	£7.95
Micromite Explore-28	07108191	£6.95
Ultrabrite LED Driver	16109191	£6.95
AUGUST 2020		
Micromite LCD BackPack V3	07106191	£9.95
Steering Wheel Audio Button to Infrared Adaptor	05105191	£7.95
JULY 2020		
AM/FM/CW Scanning HF/VHF RF Signal Generator	04106191	£13.95
Speech Synthesiser with the Raspberry Pi Zero	01106191	£5.95
PE Mini-organ PCB	AO-0720-01	£14.95
PE Mini-organ selected parts	AO-0720-02	£8.95
High-current 12V Battery Isolator – control	05106191	£6.95
High-current 12V Battery Isolator FET (2oz)	05106192	£9.95
JUNE 2020		
Arduino breakout board – 3.5-inch LCD Display	24111181	£6.95
Six-input Audio Selector main board	01110191	£10.95
Six-input Audio Selector switch panel board	01110192	
MAY 2020		
Ultra-low-distortion Preamplifier Input Selector	01111112	£11.25
Ultra-low-distortion Preamplifier pushbutton board	01111113	
Universal Regulator	18103111	£7.95
433MHz Wireless Data Repeater	15004191	£8.50
Bridge-mode Adaptor for Amplifier	01105191	£7.95
iCEstick VGA Terminal	02103191	£5.95
Analogue noise with tilt control	AO-0520-01	£7.95
Audio Spectrum Analyser	PM-0520-01	£8.95
APRIL 2020		
Flip-dot Display black coil board	19111181	£14.95
Flip-dot Display black pixels	19111182	
Flip-dot Display black frame	19111183	
Flip-dot Display green driver board	19111184	
MARCH 2020		
Diode Curve Plotter	04112181	£10.95
Steam Train Whistle / Diesel Horn Sound Generator	09106181	£8.50
Universal Passive Crossover (one off)	UPC0320	£12.50

Silicon Chip kits & parts for projects in this issue (posted from Australia):

Microphone Preamplifier short-form kit (SC6784)	https://siliconchip.au/Shop/20/6784	AUD $84.00 (about £43) + P&P
USB Keyboard Adaptor for VGA PicoMite short-form kit (SC6861)	https://siliconchip.au/Shop/20/6861	AUD $36.00 (about £18) + P&P
ps2x2pico PS/2 Adaptor kit (SC6864)	https://siliconchip.au/Shop/20/6864	AUD $39.00 (about £20) + P&P
1.5m-long 6-pin mini-DIN to mini-DIN cable	https://siliconchip.au/Shop/7/6869	AUD $12.00 (about £6) + P&P

20% VAT is included in the above prices for customers in the UK. Postage & packing cost is typically around £12-15.
Please allow 1-2 weeks for delivery (longer in some cases).

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value.

The rate for semi-display space is £10 per centimetre high, with a minimum height of 2.5cm. All semi-display advertisements have a width of 5.5cm. The prepaid rate for classified adverts is 40p per word (minimum 12 words).

For rates, further information on display and classified advertising or to book an ad, please email us at pe@pemag.au

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD
Suppliers of Electronic Components
www.bowood-electronics.co.uk
Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield,
Derbyshire S40 2QR. Sales: 01246 200 222
Send large letter stamp for Catalogue

Electrical Industries Charity (EIC)
We help people working in the electrical, electronics and energy community as well as their family members and retirees.
We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
www.electricalcharity.org
electrical industries charity

COAST ELECTRONICS
BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS
WWW.COASTELECTRONICS.CO.UK

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS?
For free stock list and/or advice, please contact me: geoffdavies337@gmail.com Telephone: 01788 574774

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits.
Visit www.cstech.co.uk

COMPONENTS AS JOB LOT
Large stock. Transistors, Diodes, Logic, Memory and Analogue parts. 1000s of passive items. Catalogued and sorted. 'New Old Stock' in racks. Email: rojobarn@gmail.com

Andrew Kenny – Qualified Patent Agent
EPO UKIPO USPTO
Circuits Electric Machinery Mechatronics
Web: www.akennypatentm.com
Email: Enquiries@akennypatentm.com
Tel: 0789 606 9725

ADVERTISING INDEX

ENIGMA COMPONENTS SHOP 51
FLOWCODE 69
HAMMOND ELECTRONICS Ltd. 13
MICROCHIP TECHNOLOGY .... Outer back cover
PCBWAY. .... Inner front cover
WIRELESS FOR THE WARRIOR BOOKS 49

Advertising mailing address
Electron Publishing Australia
PO Box 194
Matraville NSW 2036
Australia

Email pe@pemag.au
Web www.electronpublishing.com

For editorial contact details, see page 2.

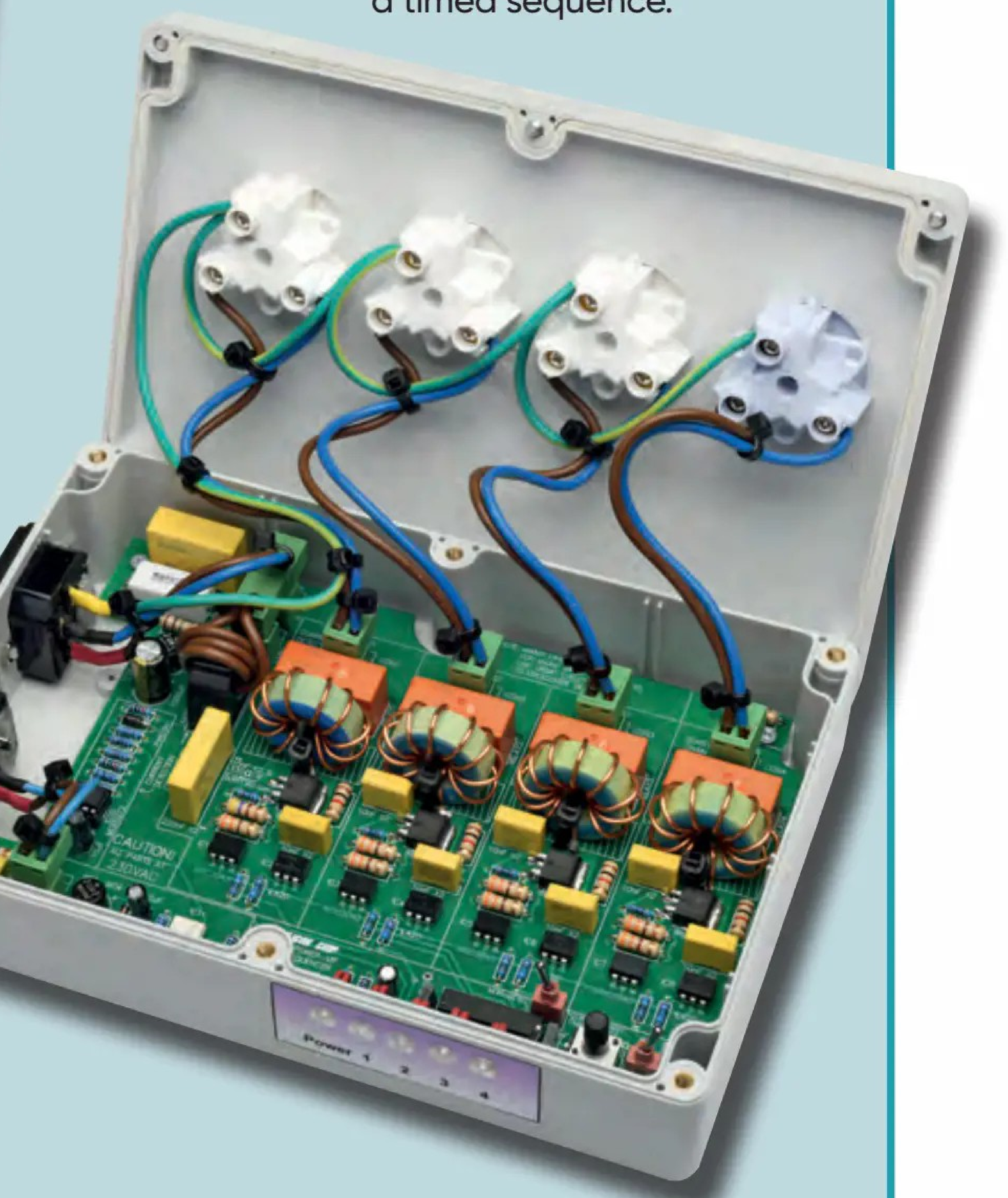
Next Month – in the February issue

Laser Communicator

A fun, educational project for transmitting sound using a laser beam.

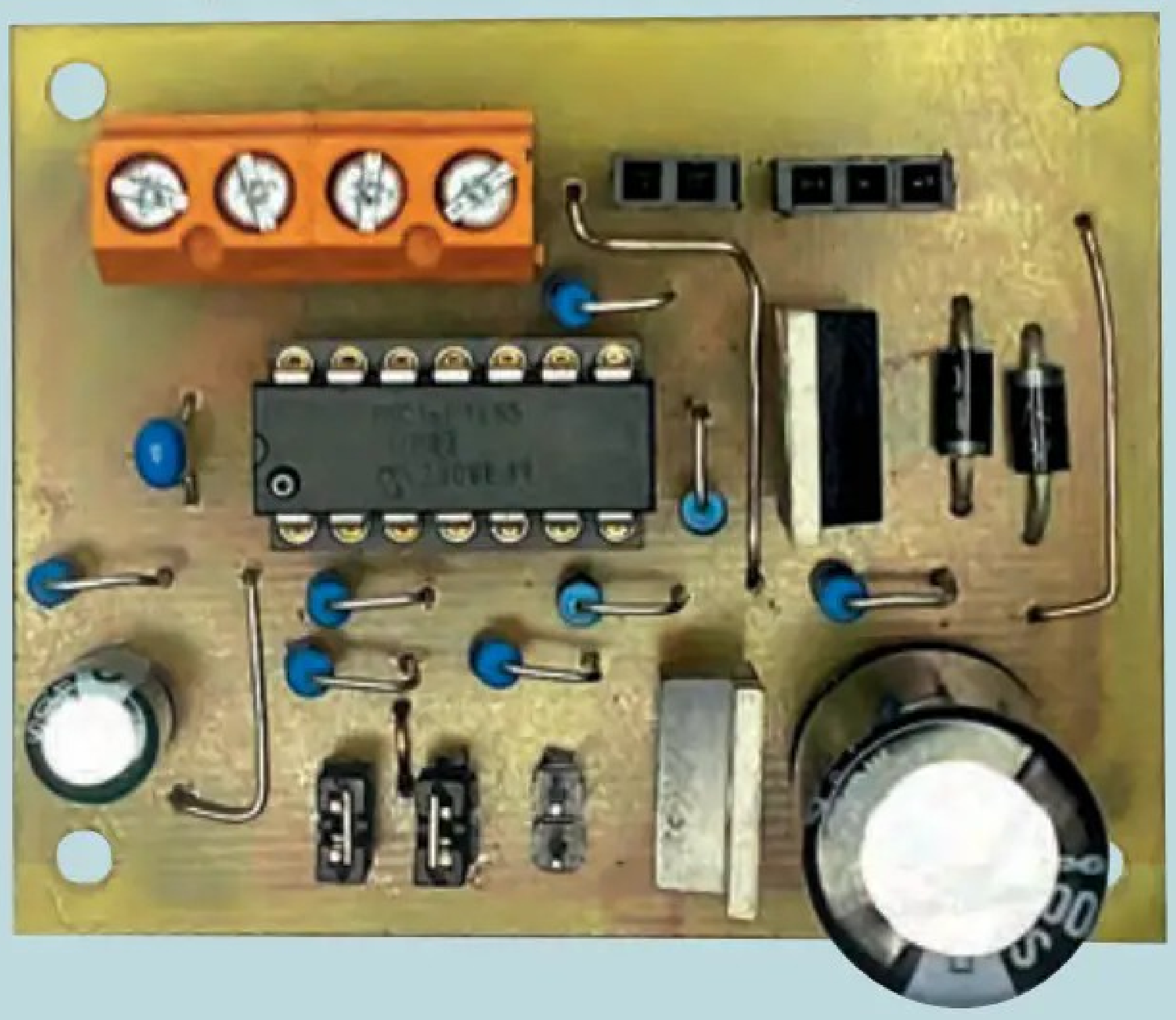
Mains power-up sequencer

Switch banks of equipment on and off in a timed sequence.



Model Railway Points Controller

Up to eight sets per controller PCB, with optional LED-based signals.



On sale 3rd of January, 2025

Content may be subject to change

Practical Electronics

Subscriptions hotline

To subscribe or renew online, paying by credit/debit card or PayPal, please visit our shop at: www.electronpublishing.com

For those wishing to pay by direct debit, cheque or over the telephone, please contact 'Select Publisher Services' by calling 01202 087631, or send a cheque (payable to: 'Practical Electronics') with your details to:

Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 01202 880299 or visit: www.electronpublishing.com

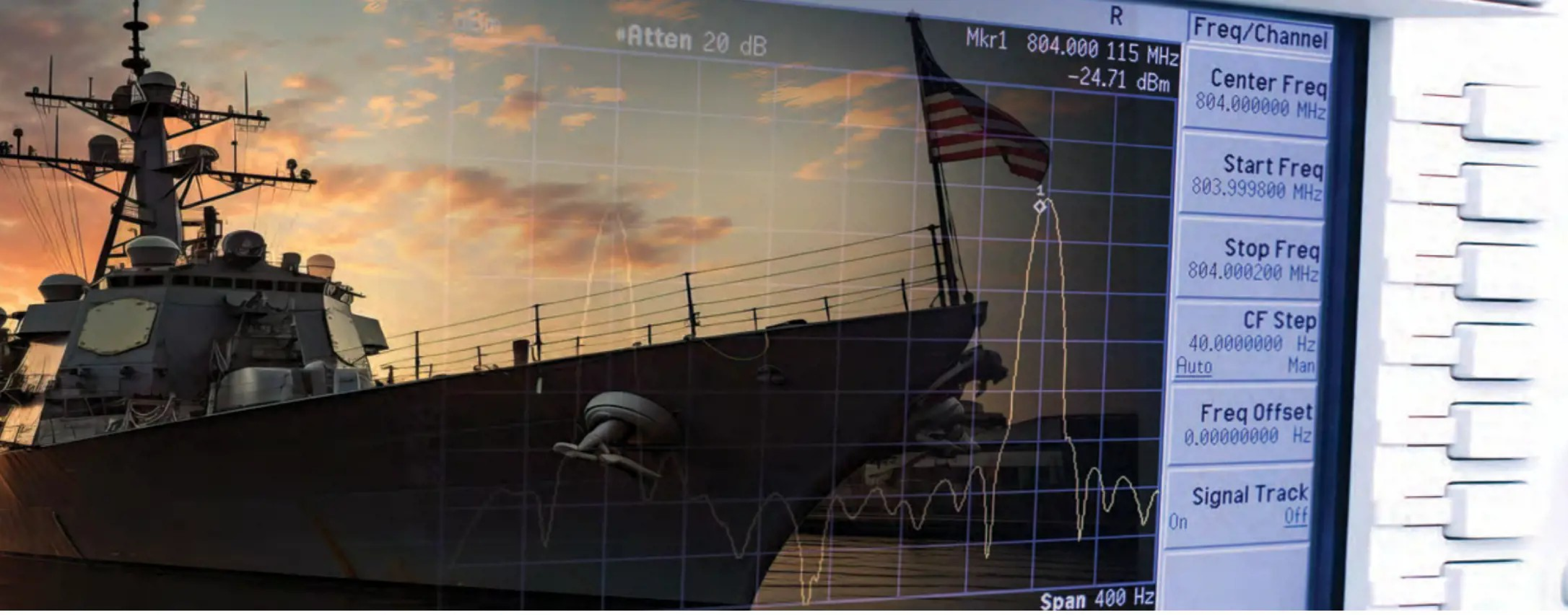
Published on approximately the first Thursday of each month by Silicon Chip Publications, U1 234 Harbord Road Brookvale NSW 2100 Australia. Printed in England by The Manson Group, Reynolds House, 8 Porters Wood, Valley Road Industrial Estate, St Albans AL3 6PZ. Distributed by Seymour, 86 Newman St, London W1T 3EX. Subscriptions UK: £33.99 (6mo); £59.99 (12mo.); £114.99 (2yrs). EUROPE: airmail service, £38.99 (6mo); £69.99 (12mo); £129.99 (2yrs). REST OF THE WORLD: airmail service, £46.99 (6mo); £84.99 (12mo); £164.99 (2yrs). Make payments to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the conditions that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter.

Did you know our online shop now sells the current issue of *PE* for £5.99 inc. p&p?

Practical Electronics
The UK's premier electronics, computing and maker magazine
Circuit Surgery
Signal filtering in the digital domain
The History of Electronics
The discoveries behind modern technology
Precision Electronics
How to make precise and repeatable measurements
Use a USB keyboard & mouse with PS/2 compatible devices
A high-quality microphone preamp for stage & studio work
Finishing the Secure Remote Switch
PLUS!
Techno Talk – Low-cost test equipment can be great
Cool Beans – Weird & Wonderful Arduino Projects, part 1
Net Work – The origins of Veroboard (stripboard)
Jan 2025 £5.99
www.electronpublishing.com
@practicalelec
practicalelectronics
60W DC load review
A low-cost device for testing supplies & batteries
Raspberry Pi Clock Radio & Media Player
PicoMite
Dec 2024 £5.99

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

www.electronpublishing.com


Atten 20 dB
R
Mkr1 804.000 115 MHz
-24.71 dBm
Freq/Channel
Center Freq
804.000000 MHz
Start Freq
803.999800 MHz
Stop Freq
804.000200 MHz
CF Step
40.0000000 Hz
Auto Man
Freq Offset
0.00000000 Hz
Signal Track
On Off
Span 400 Hz

Ultra-Low Phase Noise 101765 Voltage-Controlled Surface Acoustic Wave Oscillators

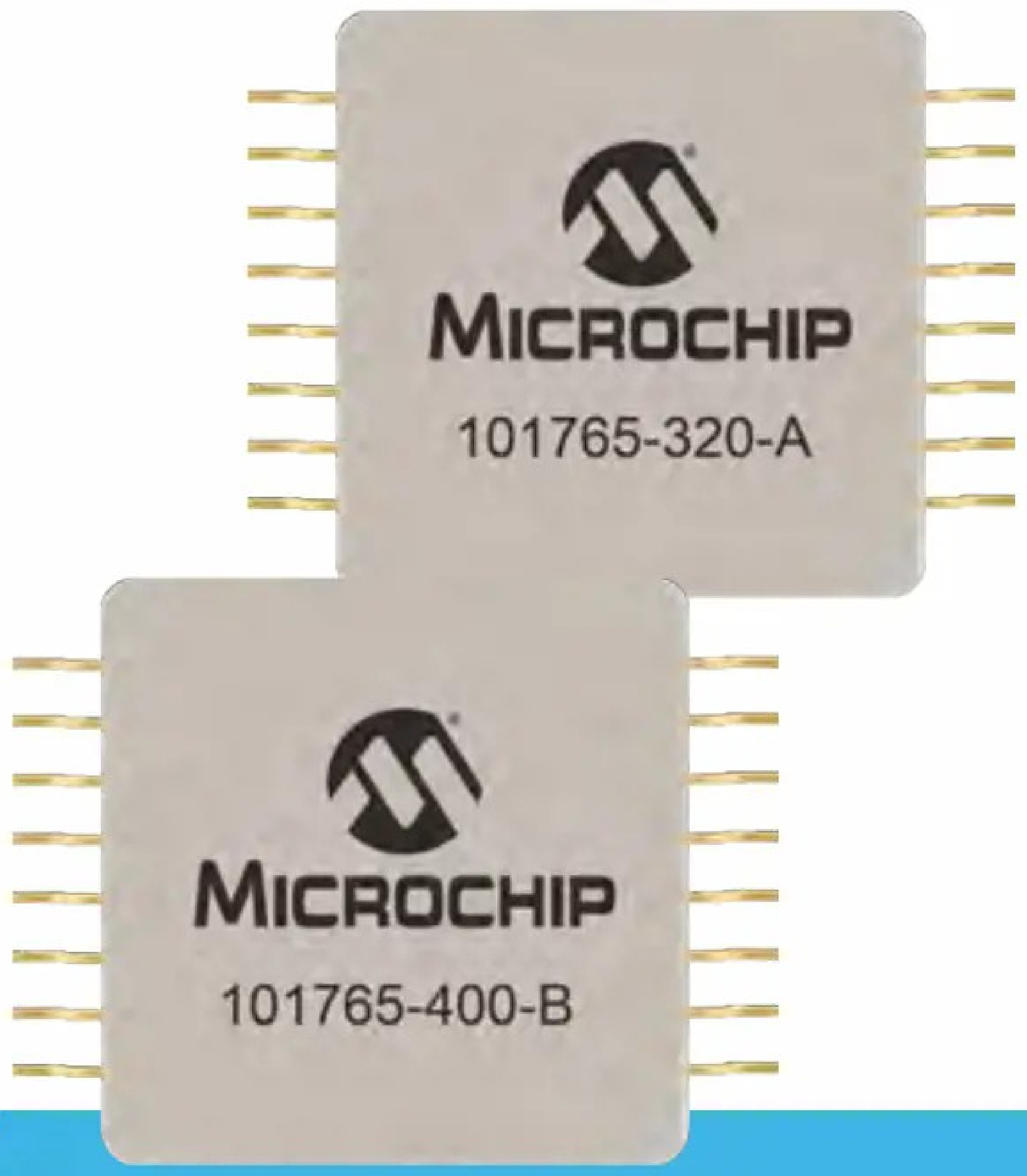
Lowest Phase Noise VCSO Per Size, Weight, Power and Cost

Introducing the 101765–320–A and 101765–400-B versions of the 101765 family, which are cutting-edge, oven-controlled, Voltage-Controlled Surface Acoustic Wave (SAW) Oscillators (VCSO) operating at 320 and 400 MHz. These high-performance oscillators leverage a high-Q quartz SAW resonator to deliver ultra-low phase noise, setting new signal precision and stability benchmarks.

Key Features

- **Ultra-low phase noise:** Exceptional signal clarity and precision
- **Stable performance:** Patented micro-oven technology ensures consistent operation across temperature ranges
- **Low-power consumption:** Energy-efficient design for longer operational life
- **Reliable and durable:** Hermetically-sealed Kovar package for enhanced protection
- **Easy integration:** Designed for surface mount installation, simplifying system integration

Elevate your technology with the world's lowest phase noise VCSOs per size, weight, power and cost (SWaP-C) and experience the next level of high-frequency performance.


MICROCHIP
101765-320-A
MICROCHIP
101765-400-B

microchip.com/101765VCSO

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2024 Microchip Technology Inc. All rights reserved.
MEC2595A-UK-11-24